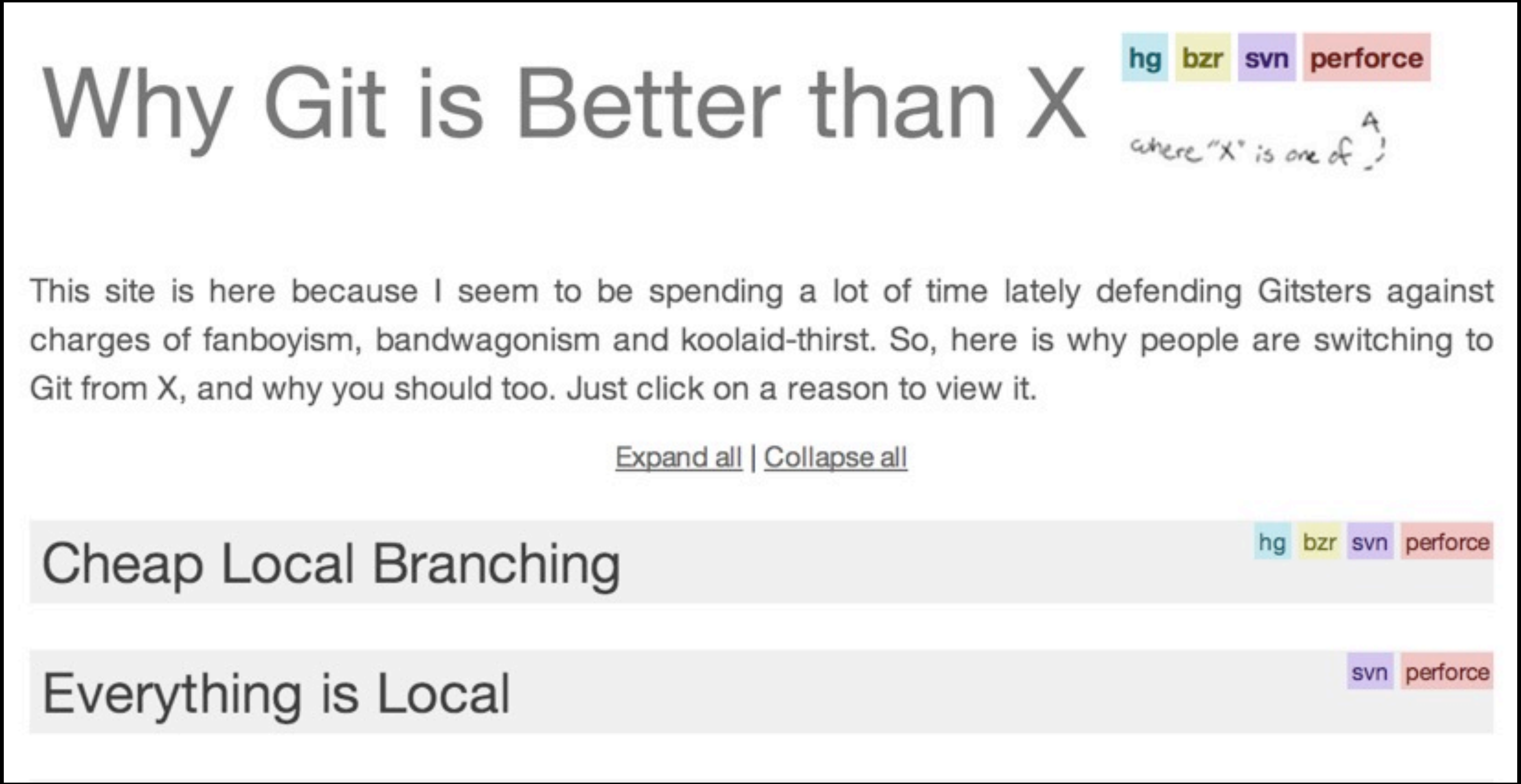
Git Introduction

For Subversion Users
Ran Tavory
Totango

Why Git?

- Because IT IS BETTER



Why Git is Better than X

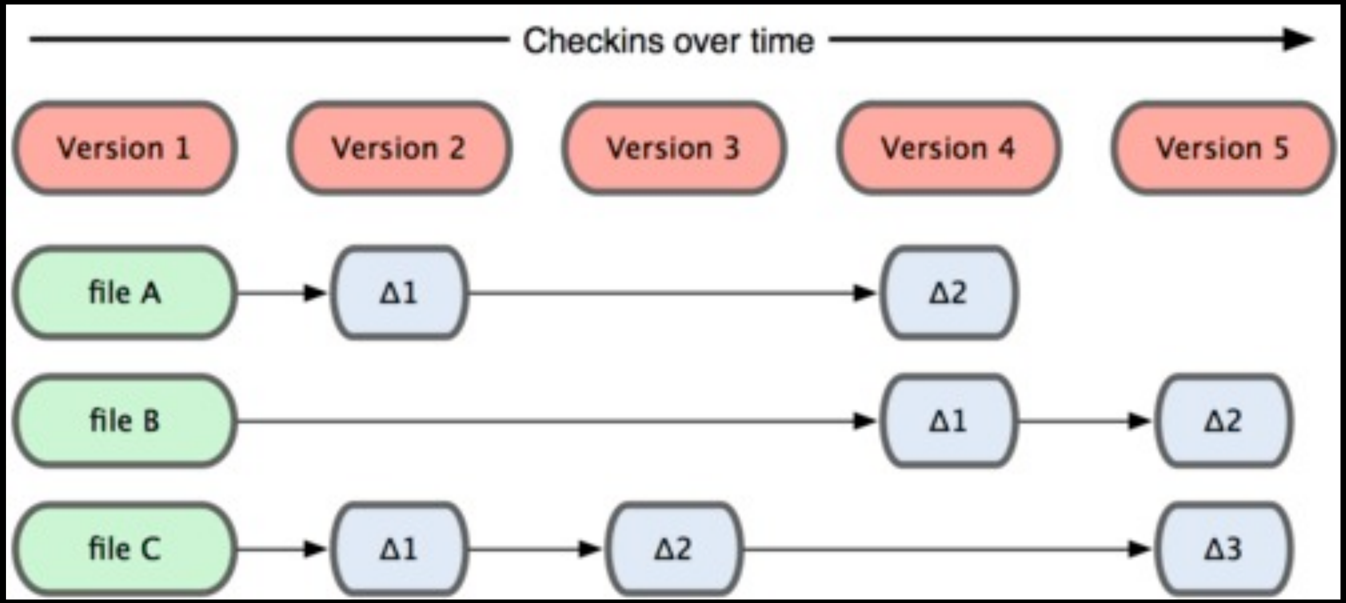
This site is here because I seem to be spending a lot of time lately defending Gitsters against charges of fanboyism, bandwagonism and koolaid-thirst. So, here is why people are switching to Git from X, and why you should too. Just click on a reason to view it.

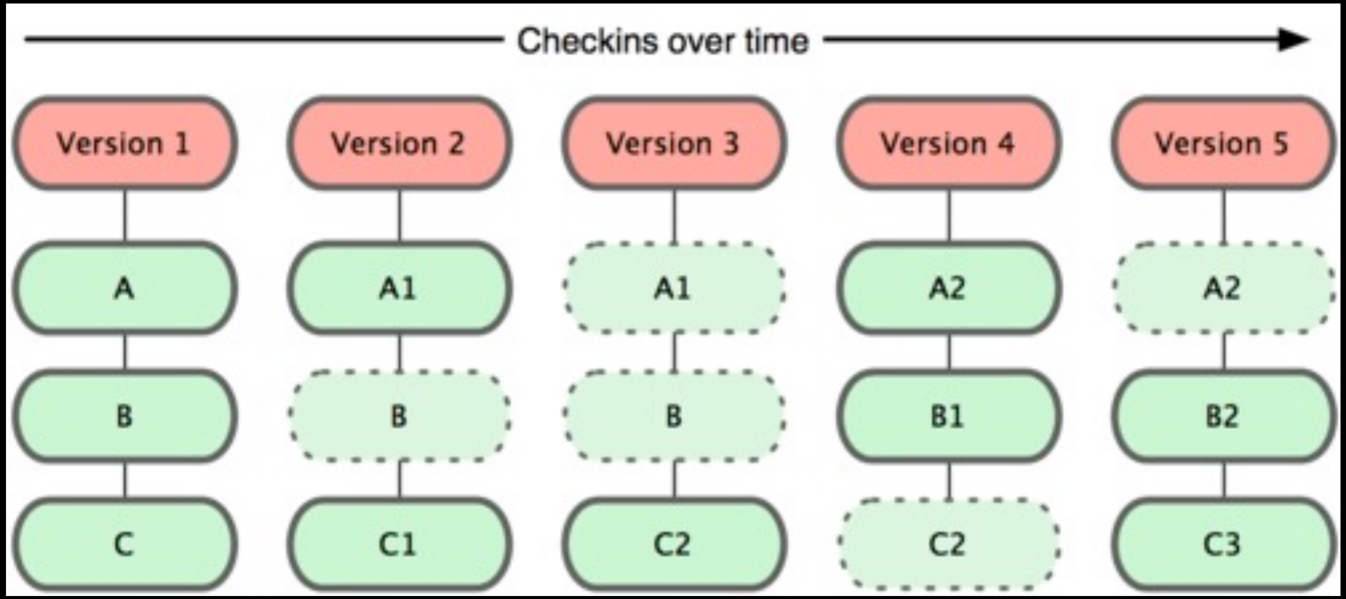
Expand all | Collapse all

Cheap Local Branching

Everything is Local

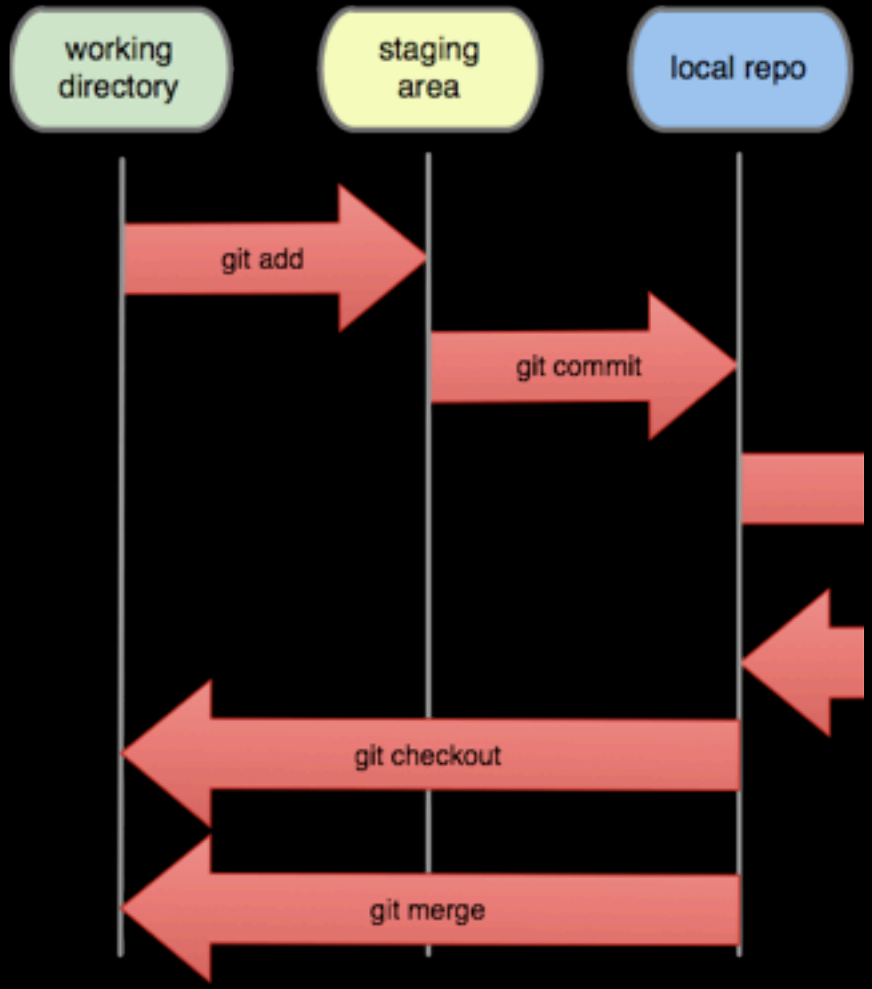
Git is a File System



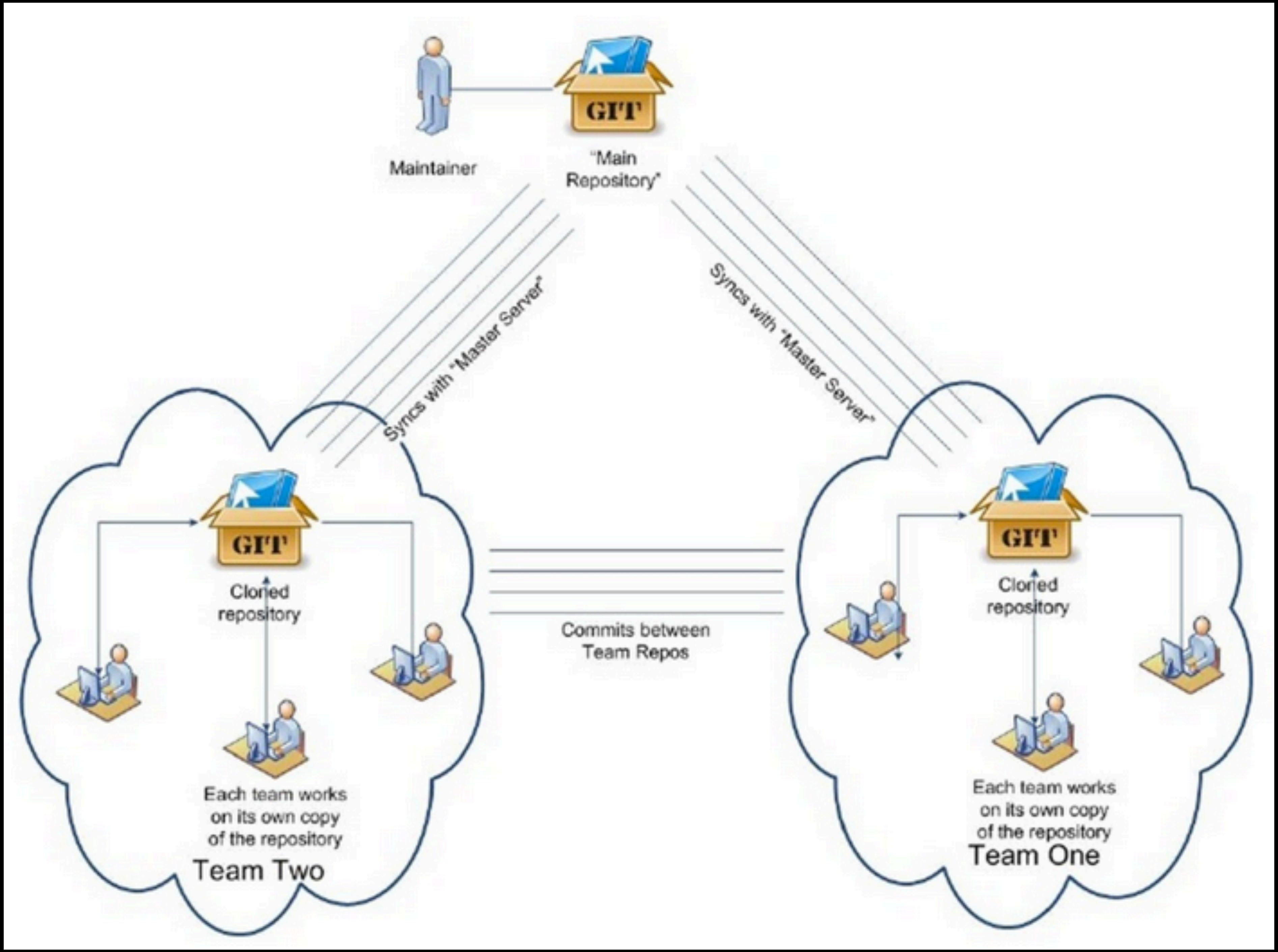


All (most) Operations are Local and Fast!

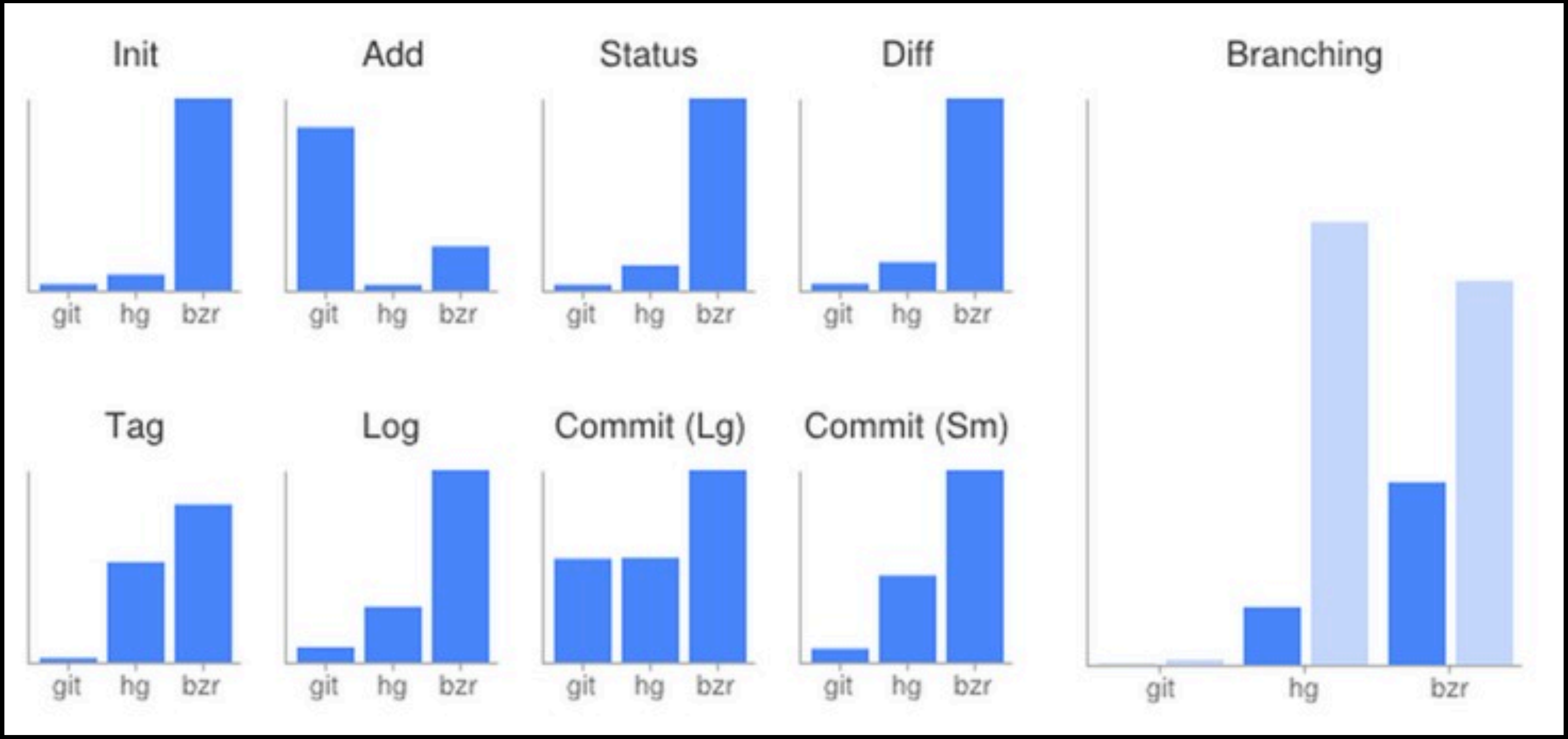
- Commit
- Log (view history)
- Diff
- ...



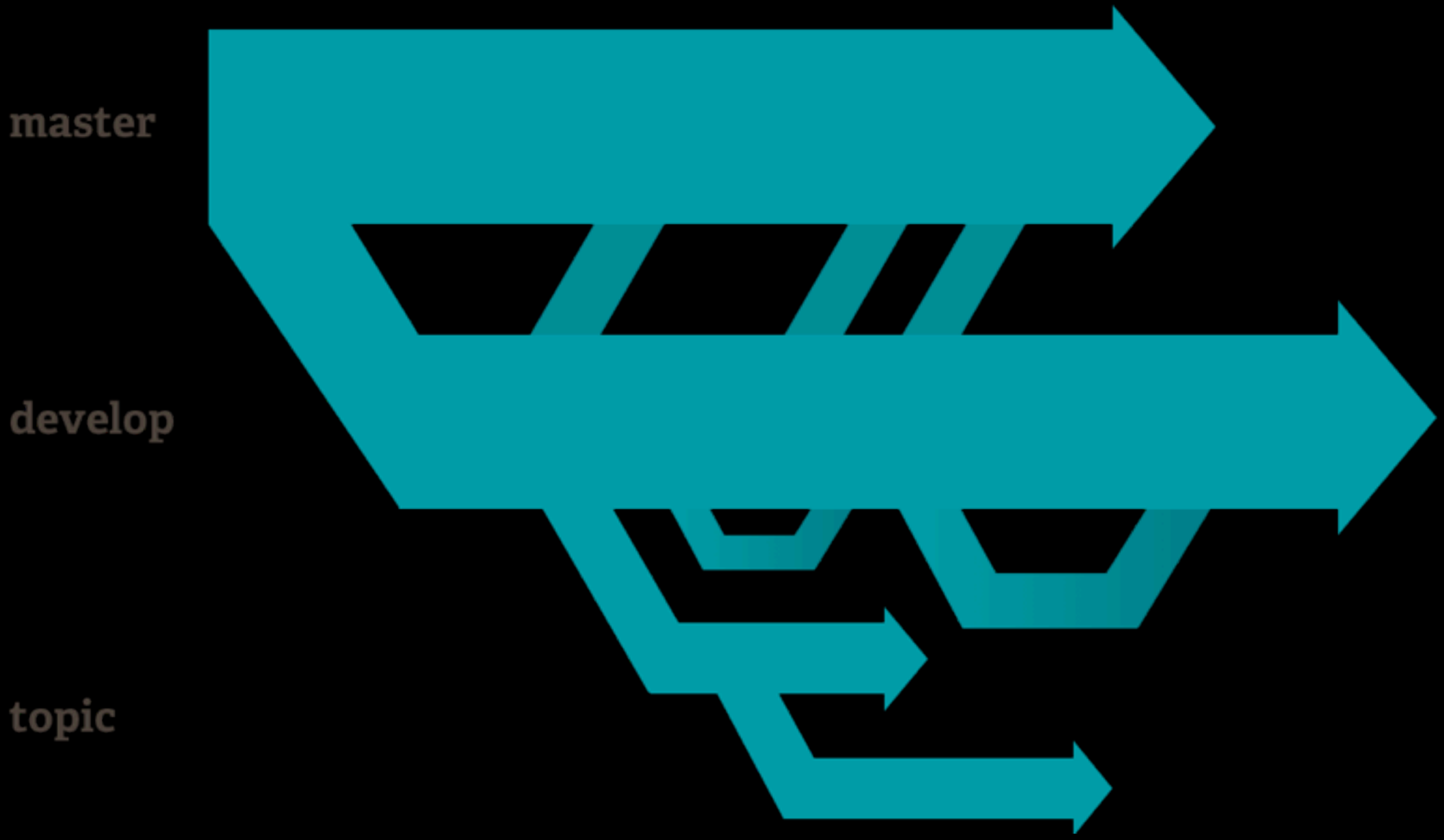
Git is Distributed


Maintainer
GIT
"Main Repository"
Syncs with "Master Server"
Syncs with "Master Server"
GIT
Cloned repository
Each team works on its own copy of the repository
Team Two
Commits between Team Repos
GIT
Cloned repository
Each team works on its own copy of the repository
Team One

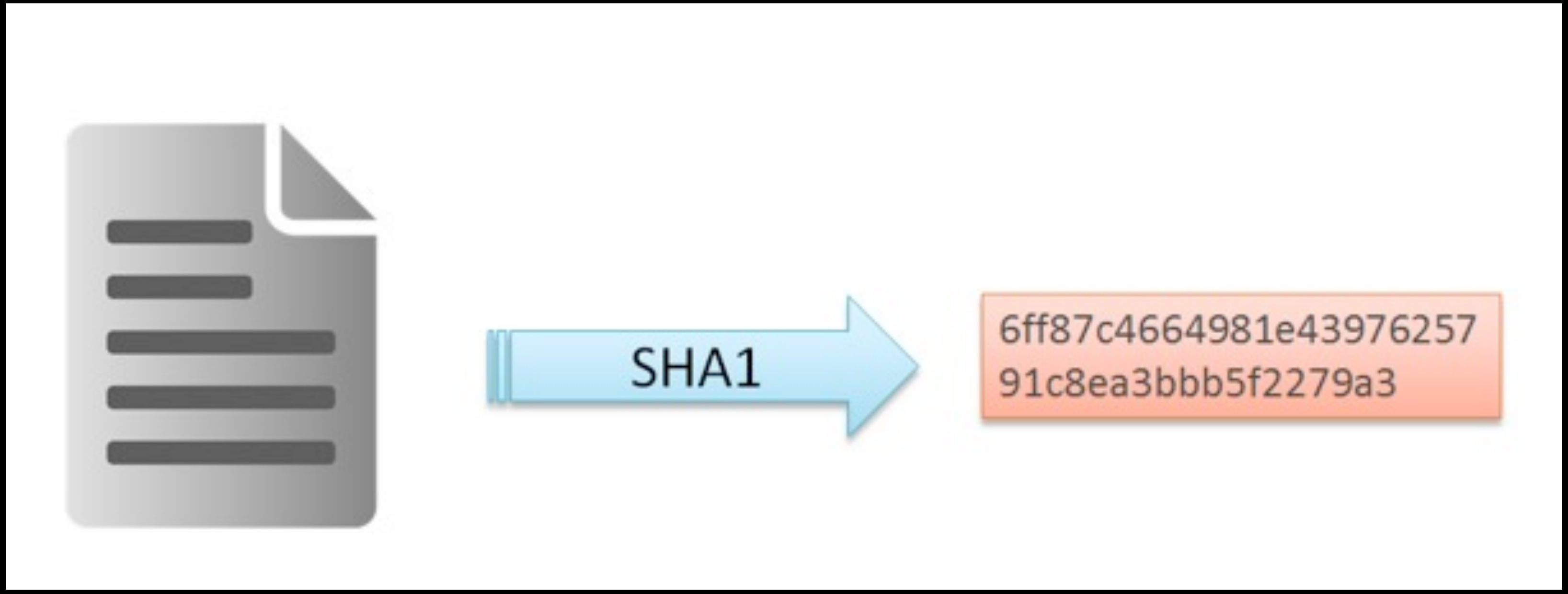
Git is Fast

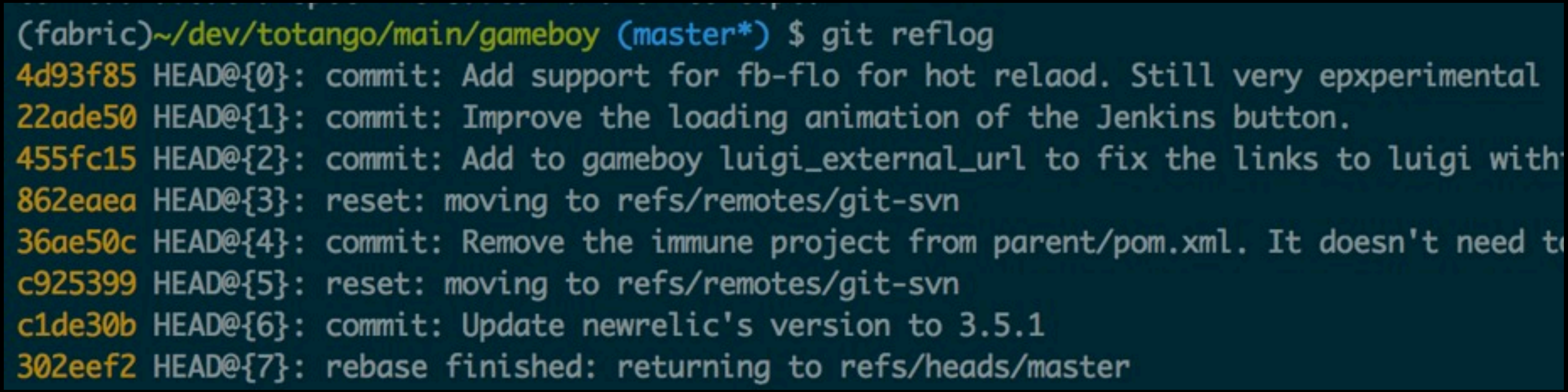

Init
Add
Status
Diff
Branching
Tag
Log
Commit (Lg)
Commit (Sm)
git
hg
bzr

Lightweight (and local) Branching



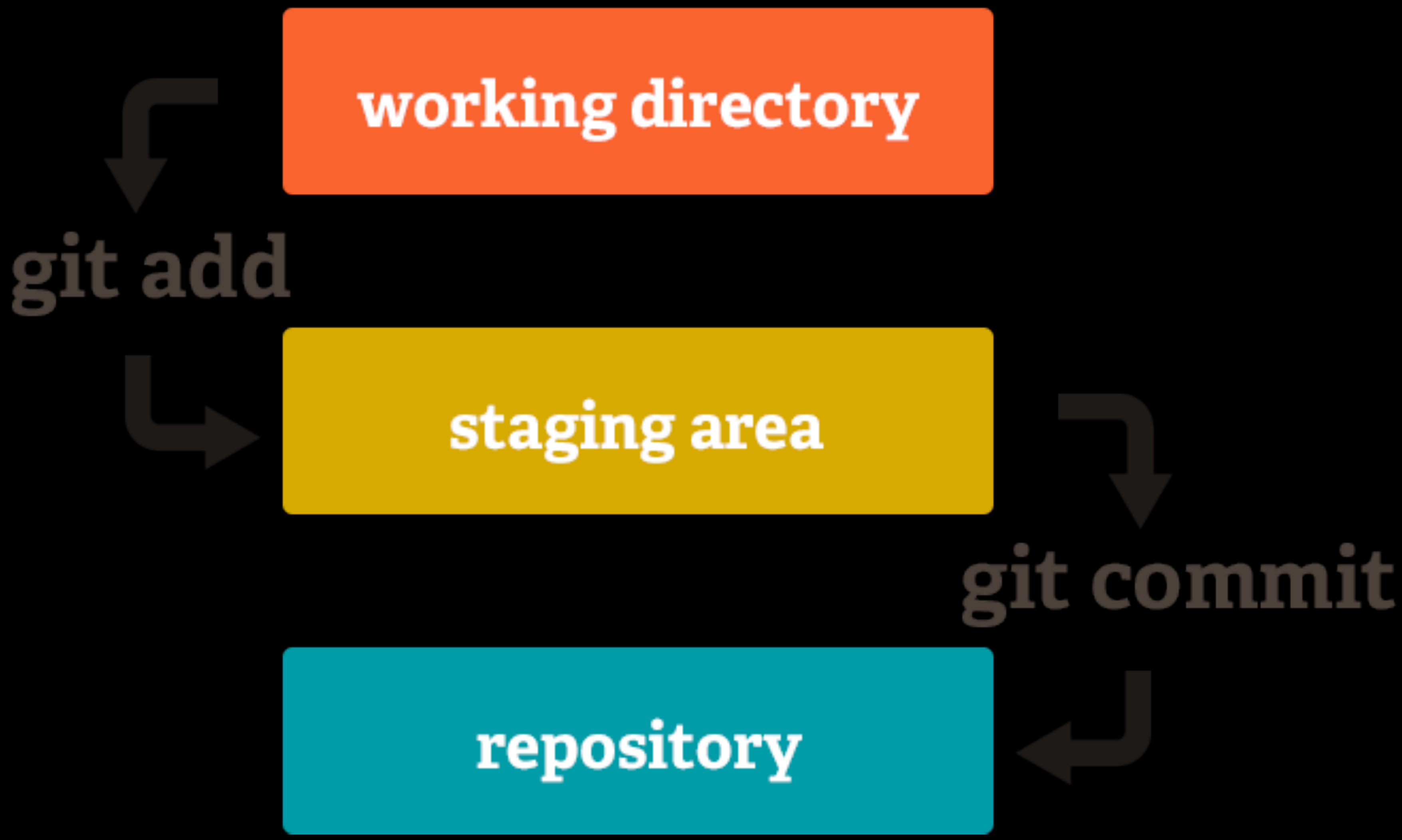
Data Integrity





```
(fabric)~/dev/totango/main/gameboy (master*) $ git reflog
4d93f85 HEAD@{0}: commit: Add support for fb-flo for hot relaod. Still very epxperimental
22ade50 HEAD@{1}: commit: Improve the loading animation of the Jenkins button.
455fc15 HEAD@{2}: commit: Add to gameboy luigi_external_url to fix the links to luigi with
862eaea HEAD@{3}: reset: moving to refs/remotes/git-svn
36ae50c HEAD@{4}: commit: Remove the immune project from parent/pom.xml. It doesn't need t
c925399 HEAD@{5}: reset: moving to refs/remotes/git-svn
c1de30b HEAD@{6}: commit: Update newrelic's version to 3.5.1
302eef2 HEAD@{7}: rebase finished: returning to refs/heads/master
```

Staging Area

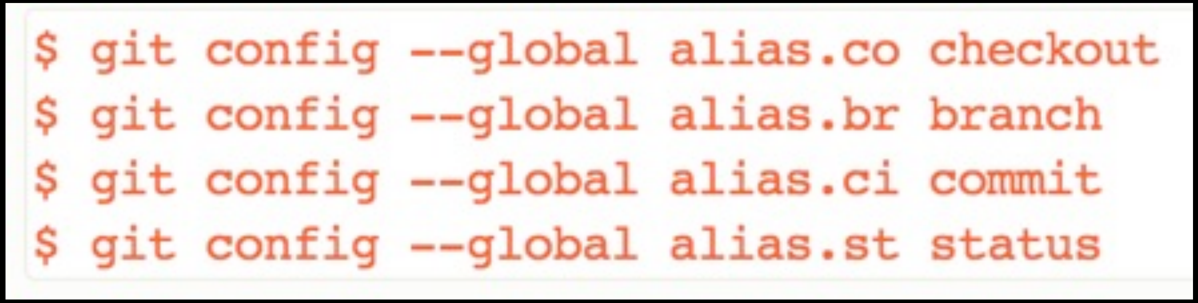


Stash Area

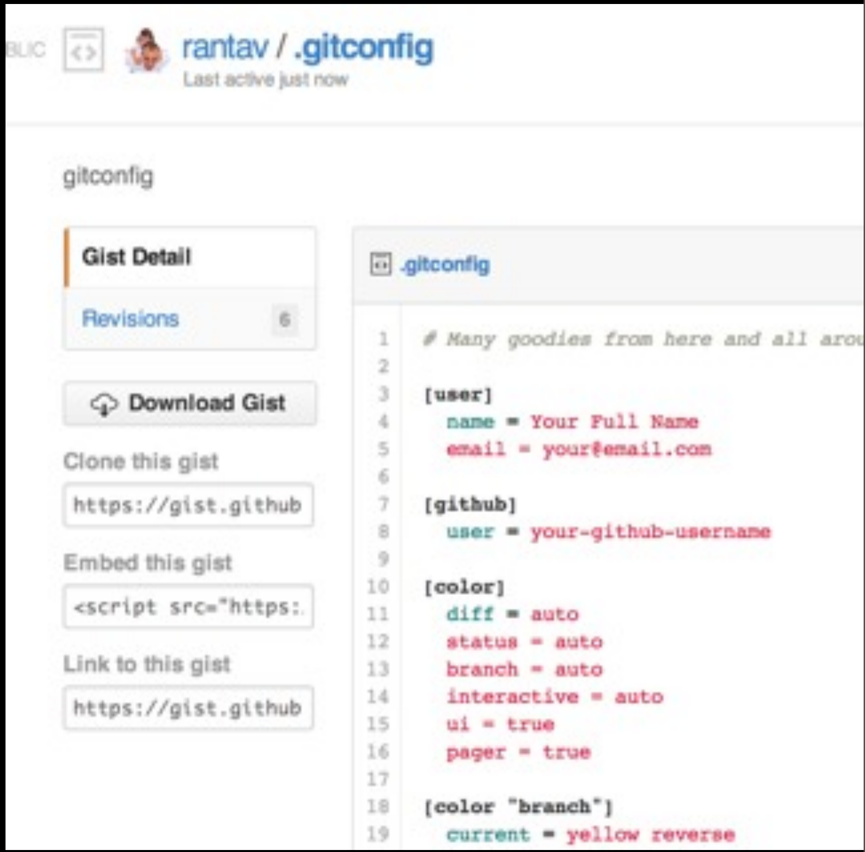
Git Setup

```
$  brew install git

$  git config --global user.name “Ran Tavory”

$  git config --global user.email “rantav@gmail.com”
```

https://gist.github.com/rantav/5d6fb14057f062ecbc70



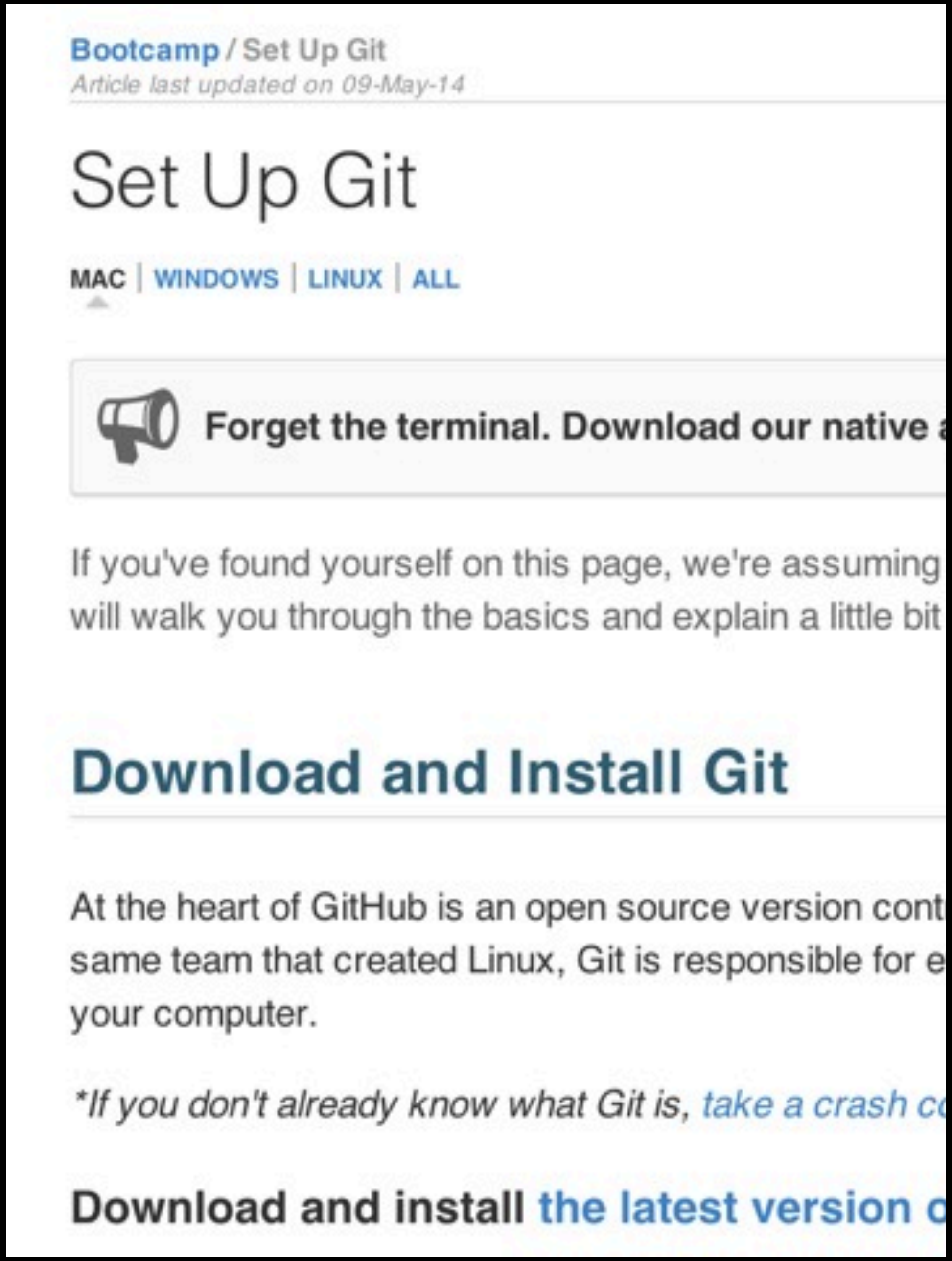
```
$ git config --global alias.co checkout
$ git config --global alias.br branch
$ git config --global alias.ci commit
$ git config --global alias.st status
```



```
# Many goodies from here and all arou

[user]
  name = Your Full Name
  email = your@email.com

[github]
  user = your-github-username

[color]
  diff = auto
  status = auto
  branch = auto
  interactive = auto
  ui = true
  pager = true

[color "branch"]
  current = yellow reverse
```

GitHub Setup

https://help.github.com/articles/set-up-git


Bootcamp / Set Up Git
Article last updated on 09-May-14

Set Up Git

MAC | WINDOWS | LINUX | ALL

Forget the terminal. Download our native

If you've found yourself on this page, we're assuming
will walk you through the basics and explain a little bit

Download and Install Git

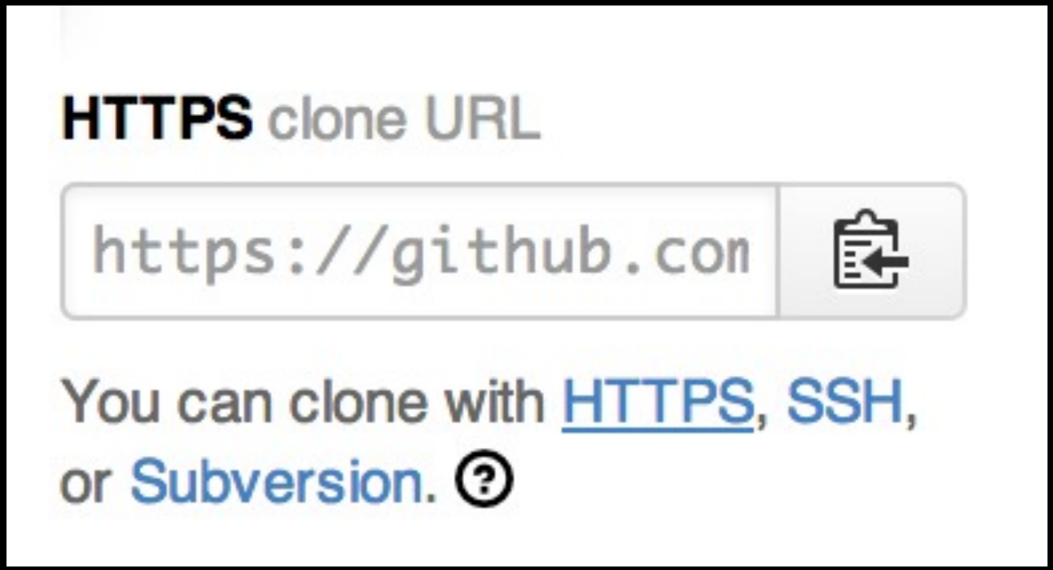
At the heart of GitHub is an open source version cont
same team that created Linux, Git is responsible for e
your computer.

*If you don't already know what Git is, take a crash c

Download and install the latest version o

Clone the Repo

```
$ git clone https://github.com/totango/main.git
```

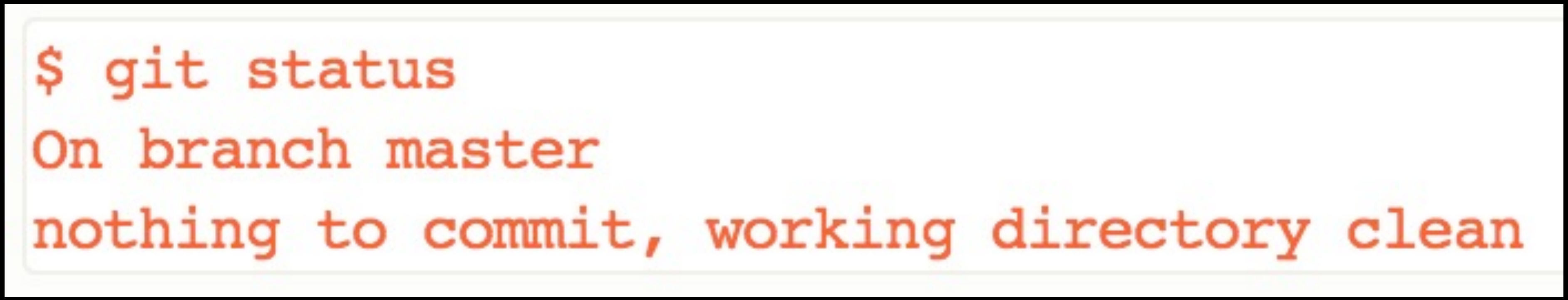


HTTPS clone URL

https://github.com

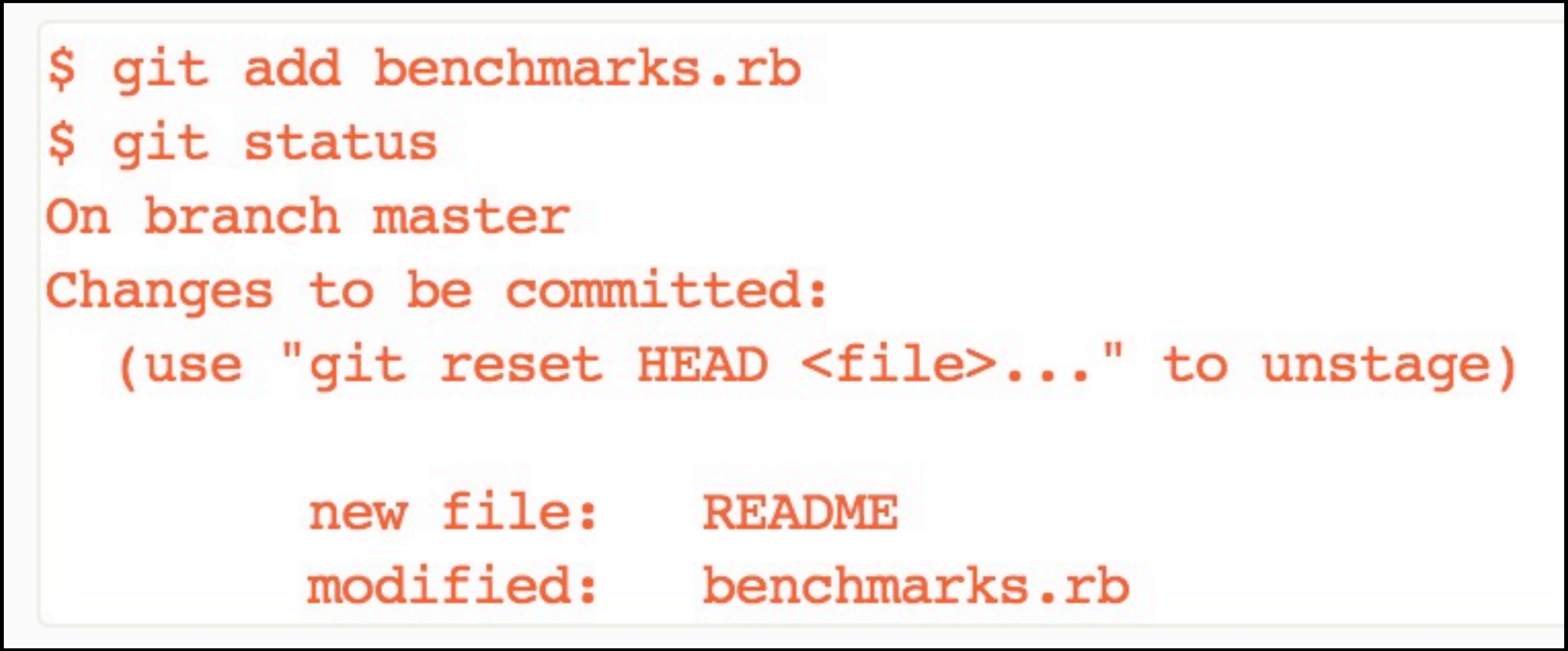
You can clone with HTTPS, SSH, or Subversion.

git status



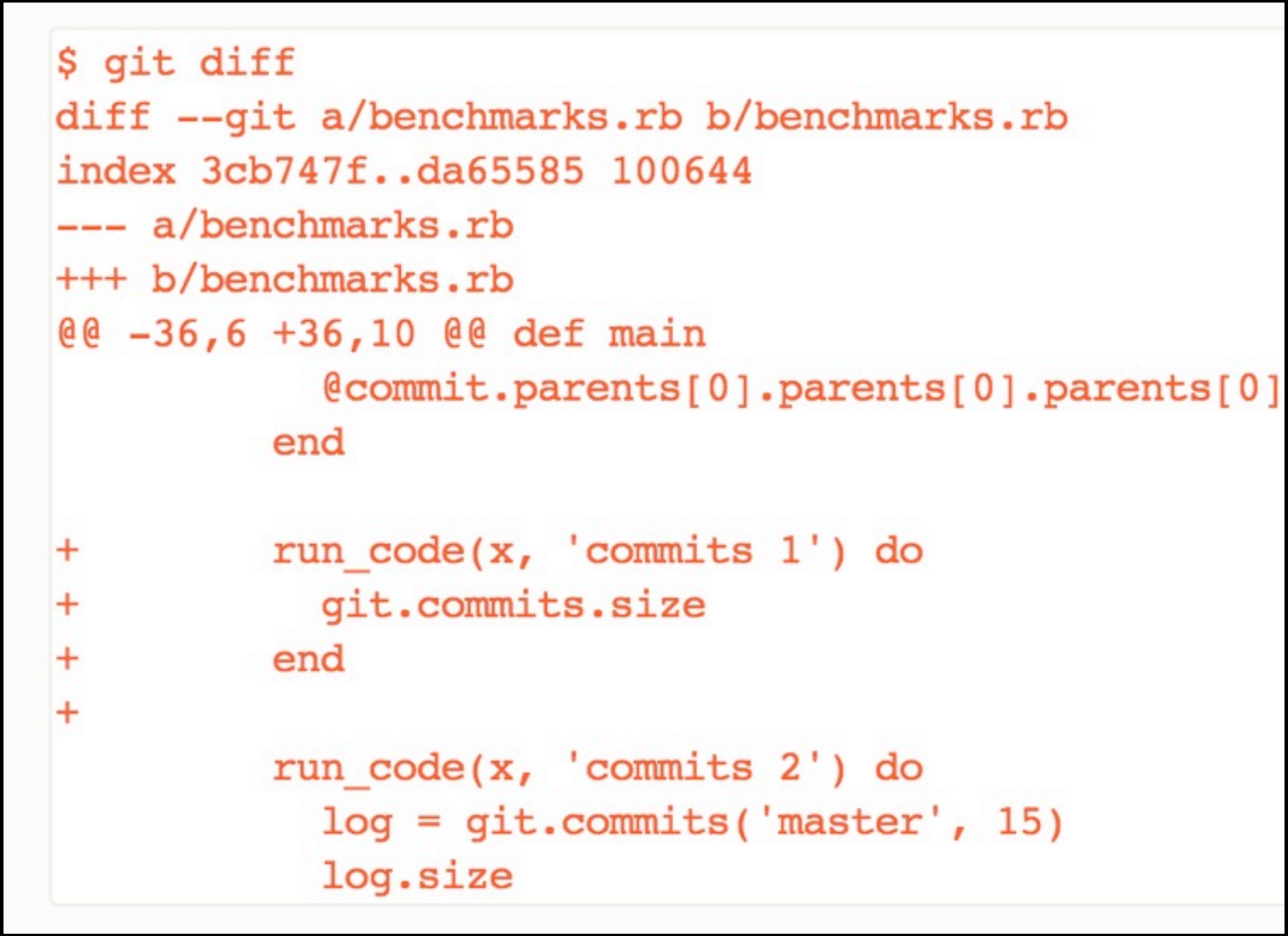
```
$ git status
On branch master
nothing to commit, working directory clean
```

git add



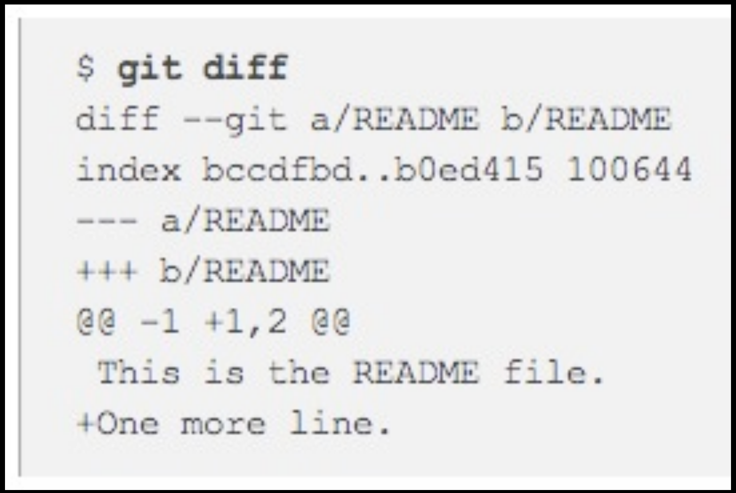
```
$ git add benchmarks.rb
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   README
        modified:   benchmarks.rb
```

git diff

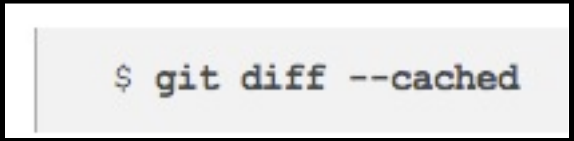


```
$ git diff
diff --git a/benchmarks.rb b/benchmarks.rb
index 3cb747f..da65585 100644
--- a/benchmarks.rb
+++ b/benchmarks.rb
@@ -36,6 +36,10 @@ def main
           @commit.parents[0].parents[0].parents[0]
         end

+        run_code(x, 'commits 1') do
+          git.commits.size
+        end
+
         run_code(x, 'commits 2') do
           log = git.commits('master', 15)
           log.size
```

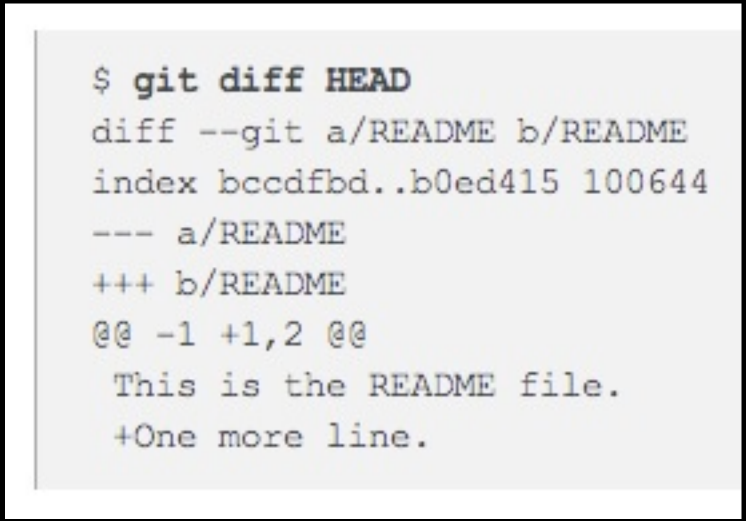
git diff



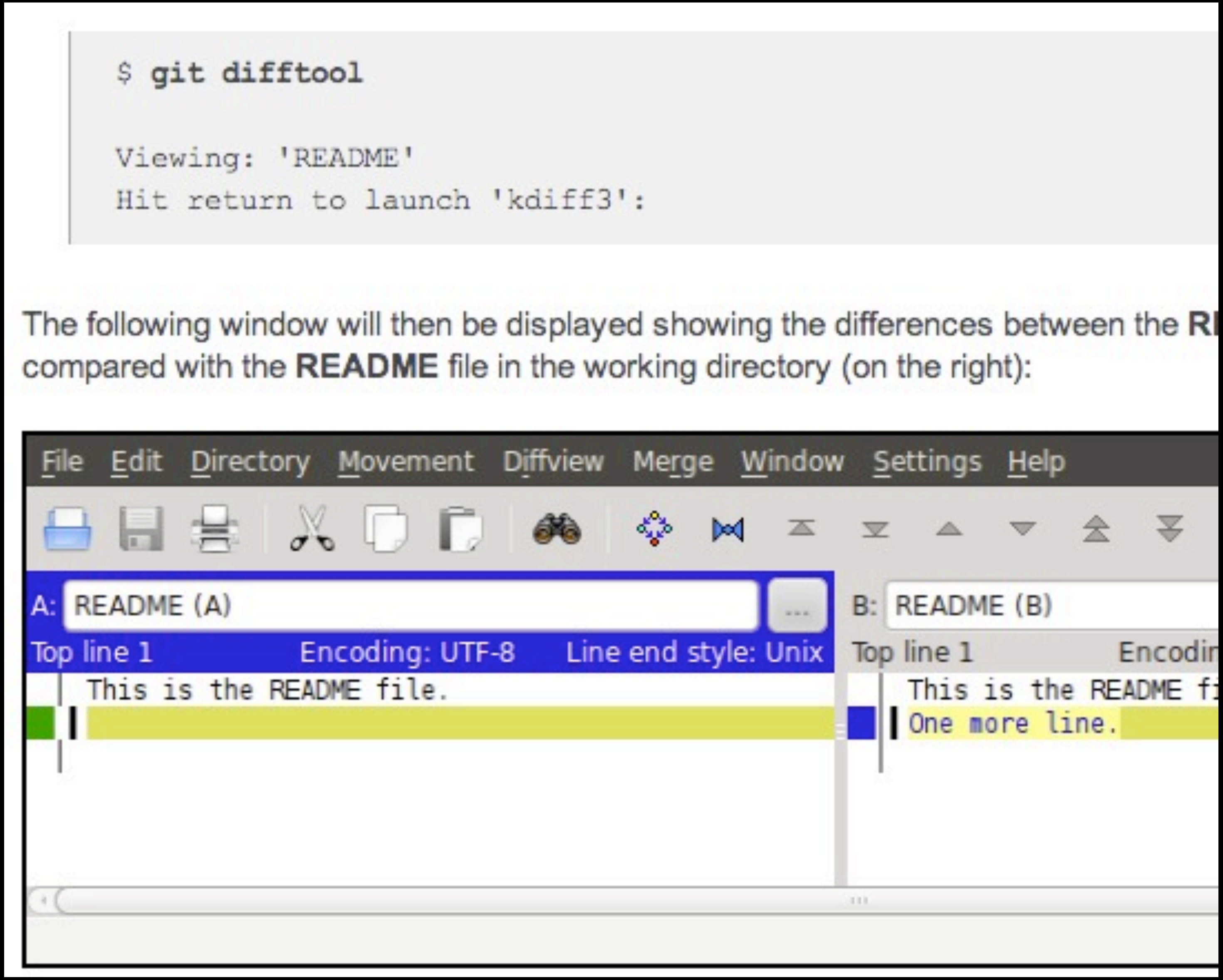
```
$ git diff
diff --git a/README b/README
index bccdfbd..b0ed415 100644
--- a/README
+++ b/README
@@ -1 +1,2 @@
 This is the README file.
+One more line.
```



```
$ git diff --cached
```



```
$ git diff HEAD
diff --git a/README b/README
index bccdfbd..b0ed415 100644
--- a/README
+++ b/README
@@ -1 +1,2 @@
 This is the README file.
 +One more line.
```



```
$ git difftool

Viewing: 'README'
Hit return to launch 'kdiff3':
```

The following window will then be displayed showing the differences between the **RI**
compared with the **README** file in the working directory (on the right):

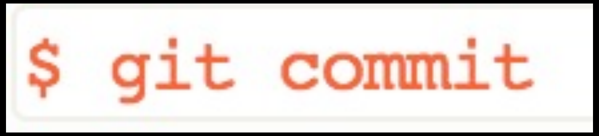
File Edit Directory Movement Diffview Merge Window Settings Help
A: README (A)
Top line 1 Encoding: UTF-8 Line end style: Unix
B: README (B)
Top line 1 Encodin

```
This is the README file.
```

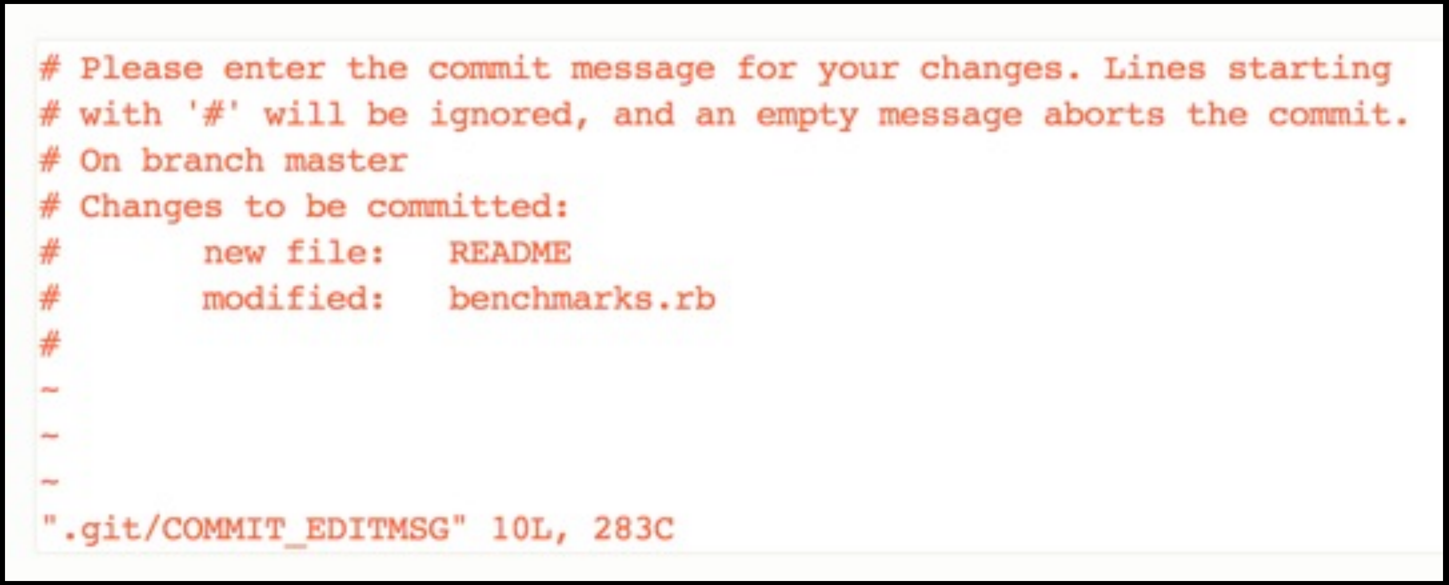
```
This is the README fi
One more line.
```

More here: http://www.gitguys.com/topics/git-diff/

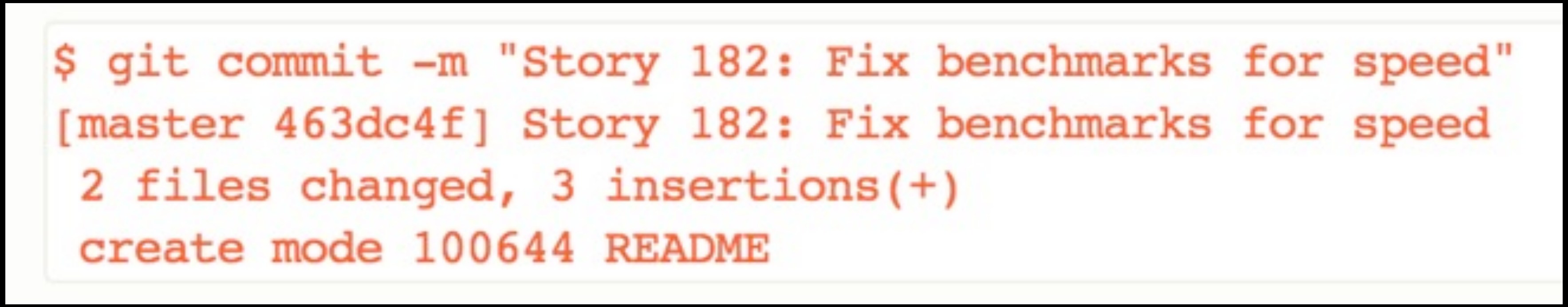
git commit



```
$ git commit
```

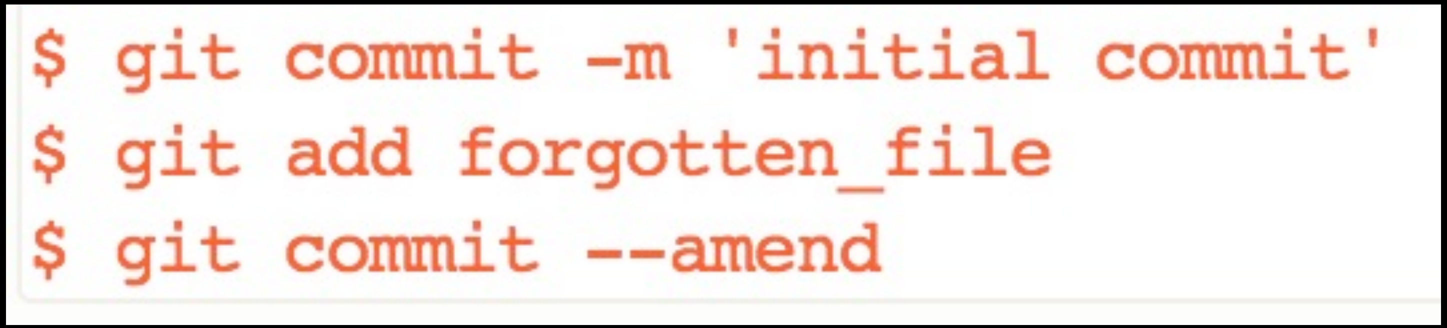


```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Changes to be committed:
#       new file:   README
#       modified:   benchmarks.rb
#
~
~
~
".git/COMMIT_EDITMSG" 10L, 283C
```



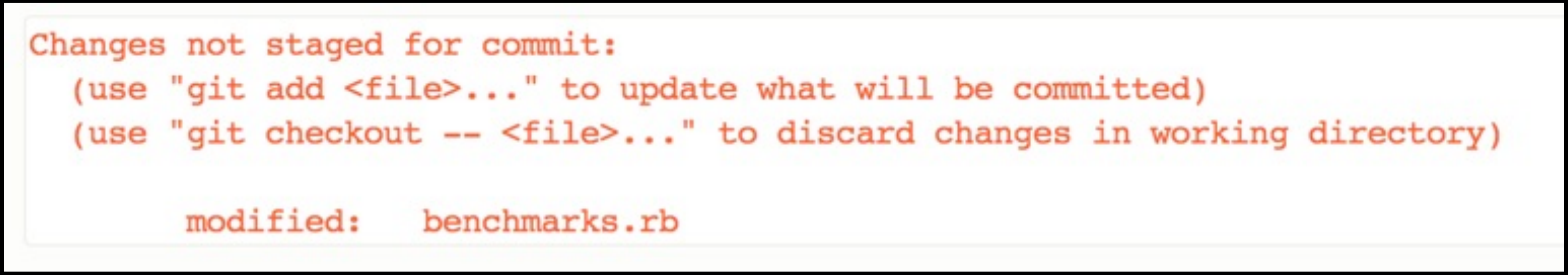
```
$ git commit -m "Story 182: Fix benchmarks for speed"
[master 463dc4f] Story 182: Fix benchmarks for speed
 2 files changed, 3 insertions(+)
 create mode 100644 README
```

git commit --amend

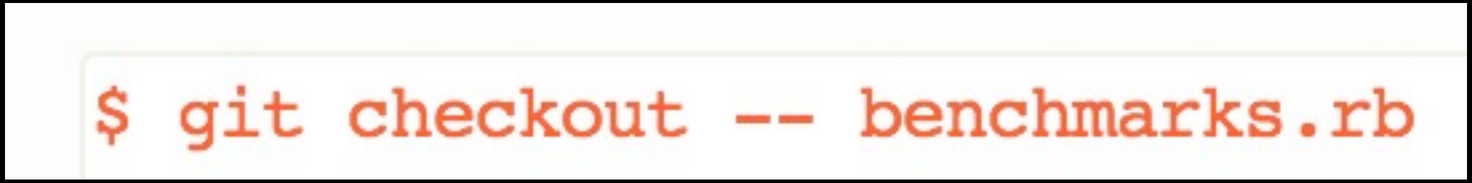


```
$ git commit -m 'initial commit'
$ git add forgotten_file
$ git commit --amend
```

git checkout

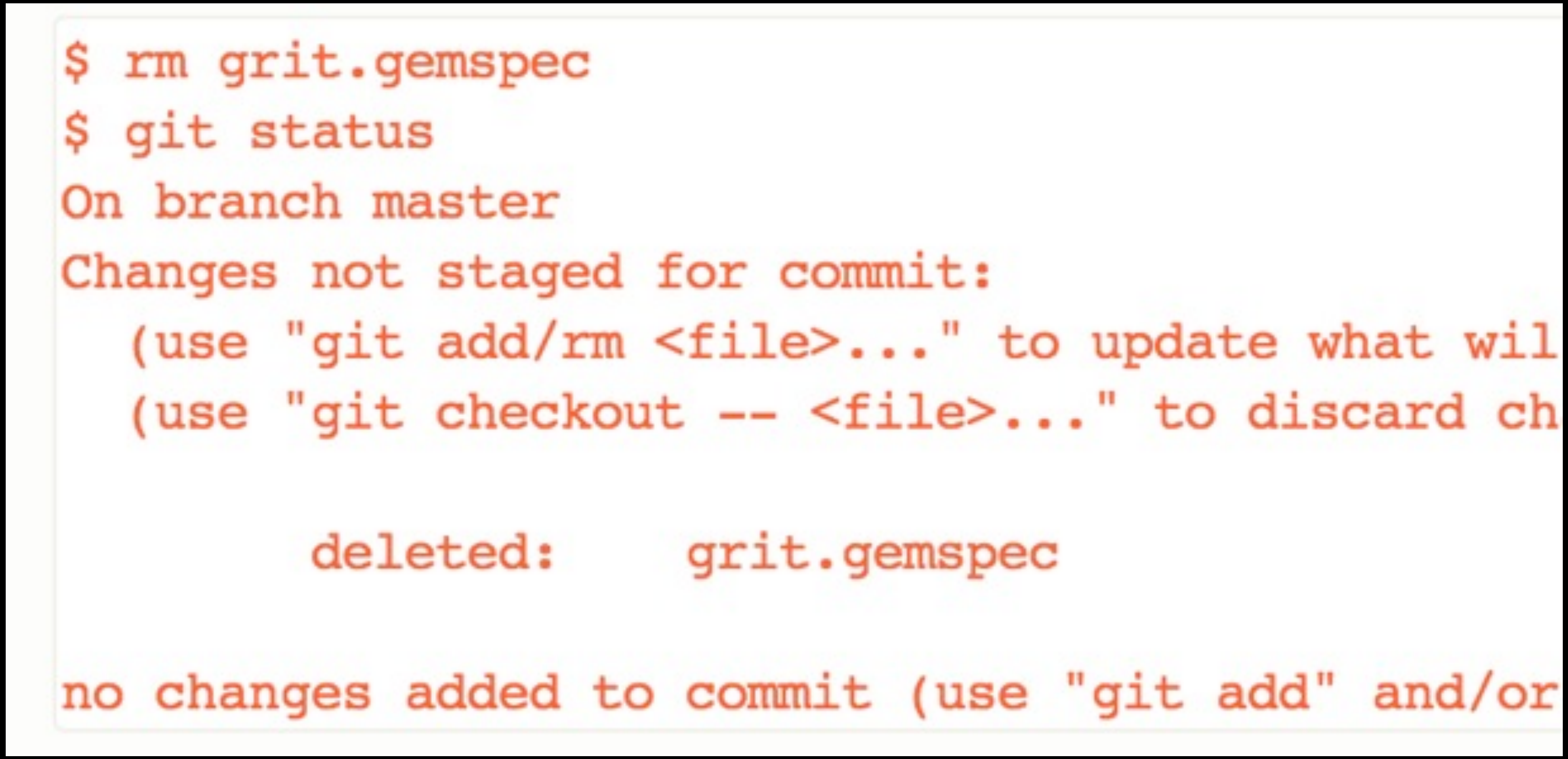


```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   benchmarks.rb
```

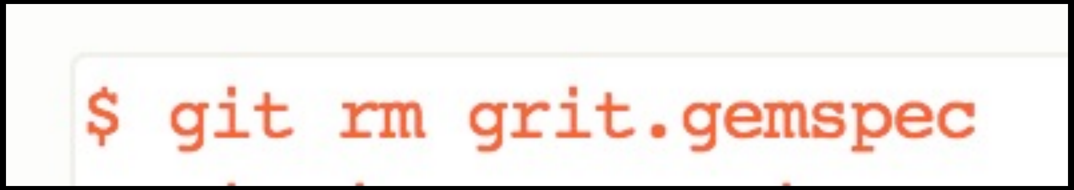


```
$ git checkout -- benchmarks.rb
```

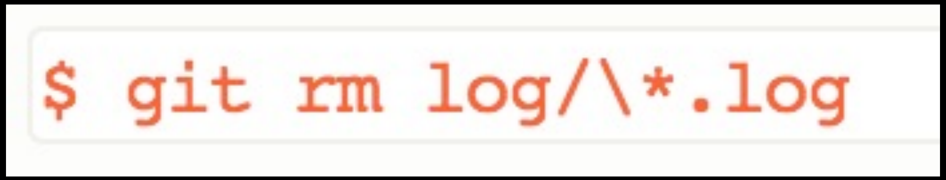
git rm



```
$ rm grit.gemspec
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what wil
  (use "git checkout -- <file>..." to discard ch

        deleted:    grit.gemspec

no changes added to commit (use "git add" and/or
```

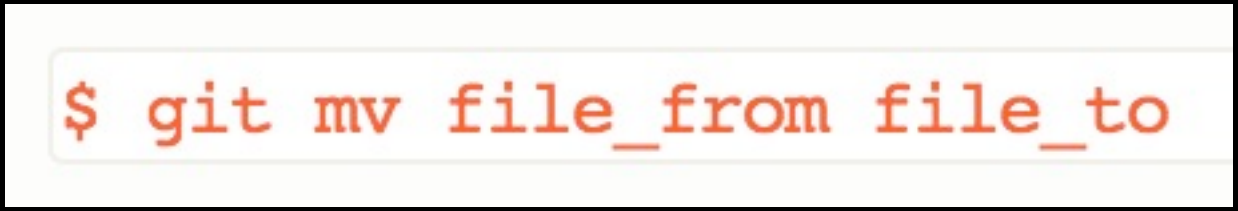


```
$ git rm grit.gemspec
```



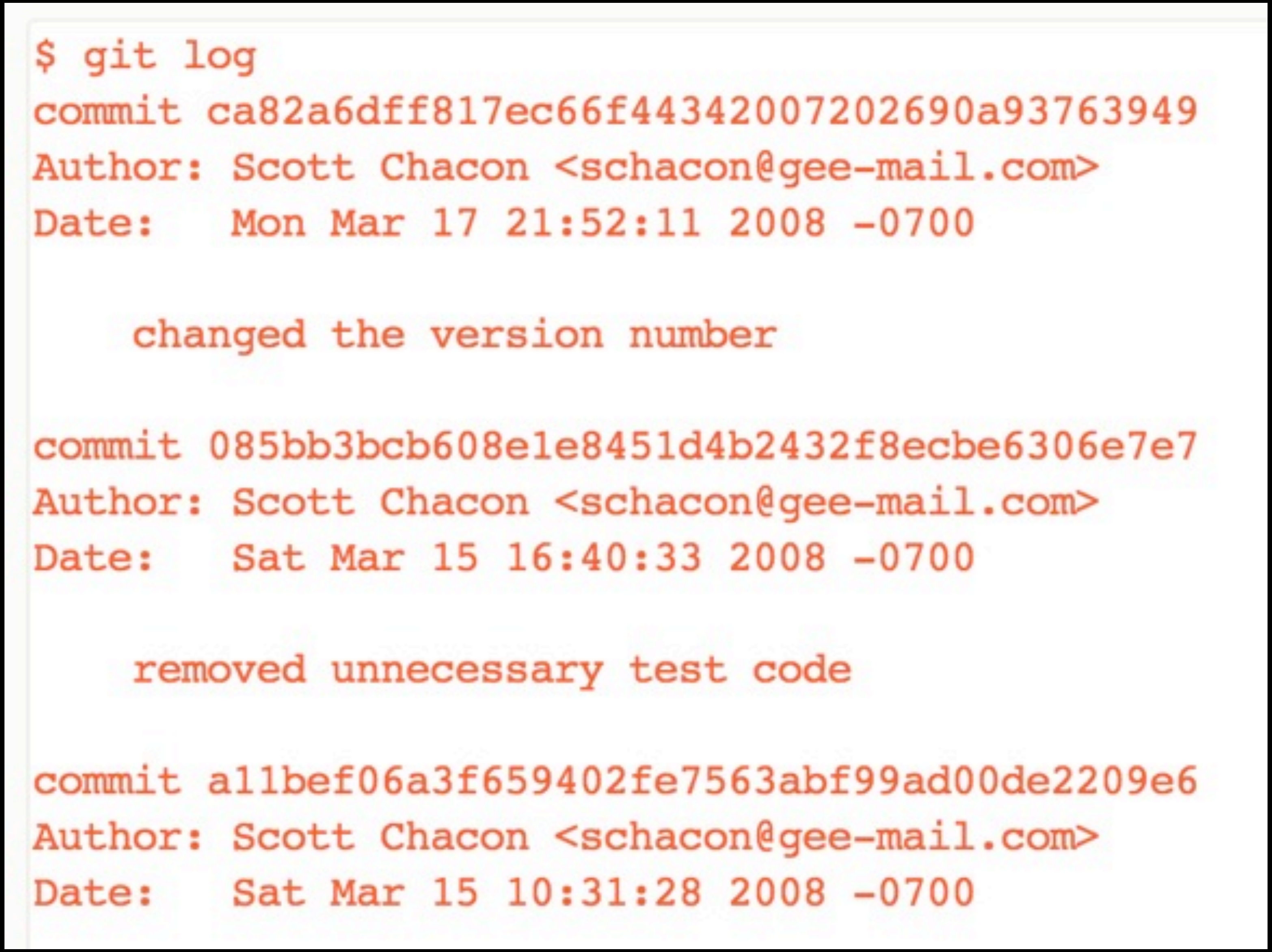
```
$ git rm log/\*.log
```

git mv



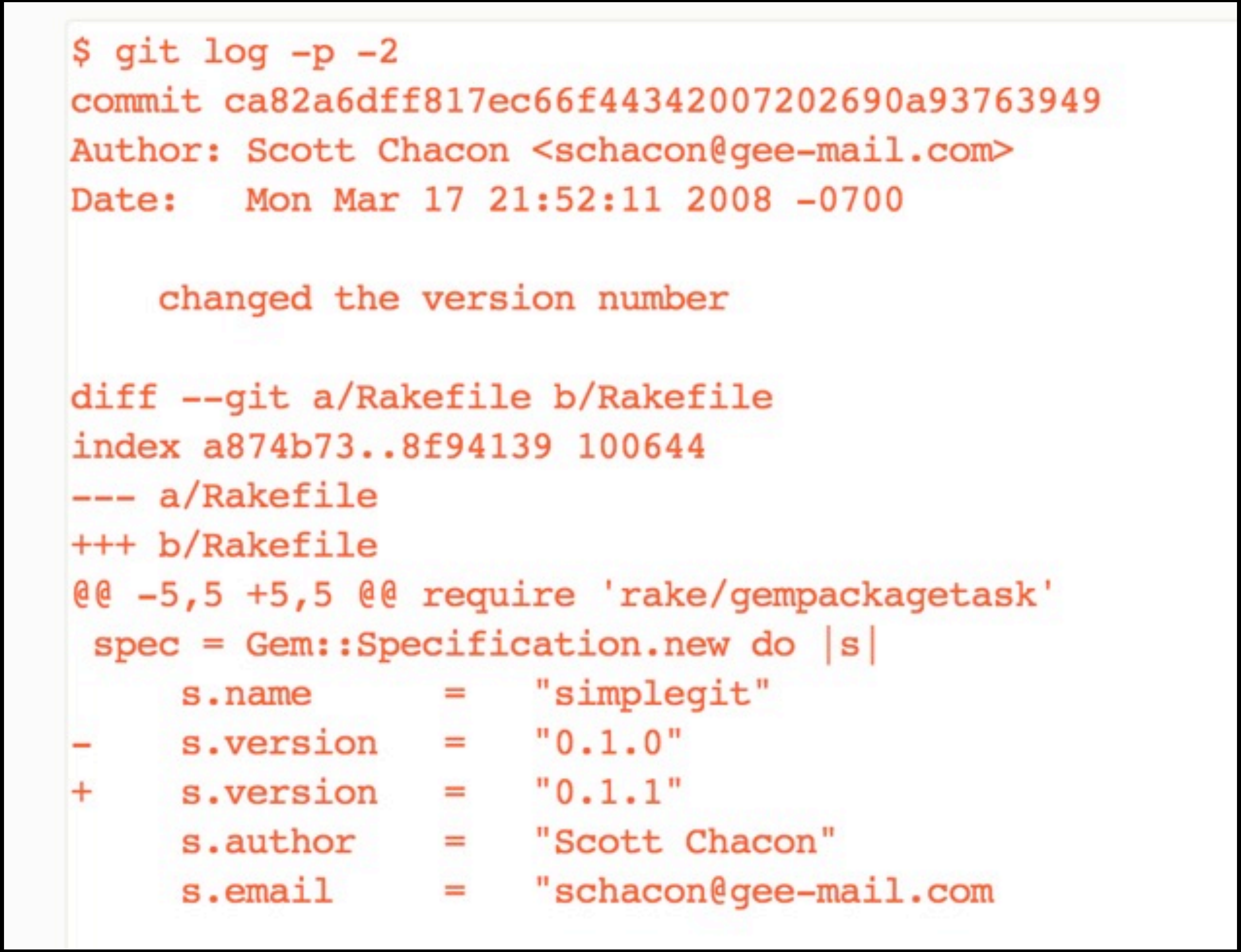
```
$ git mv file_from file_to
```

git log



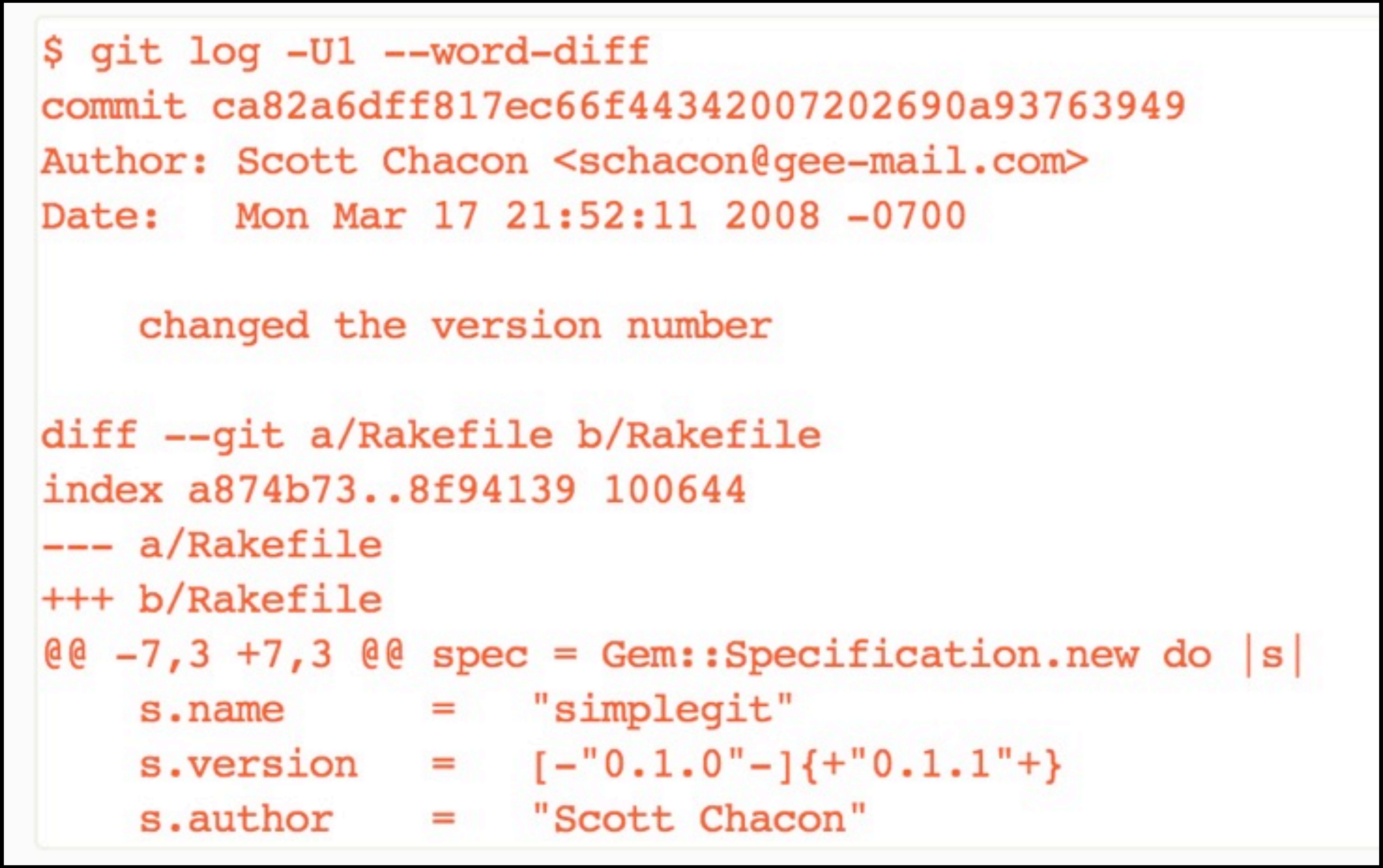
```
$ git log
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 10:31:28 2008 -0700
```

git log -p



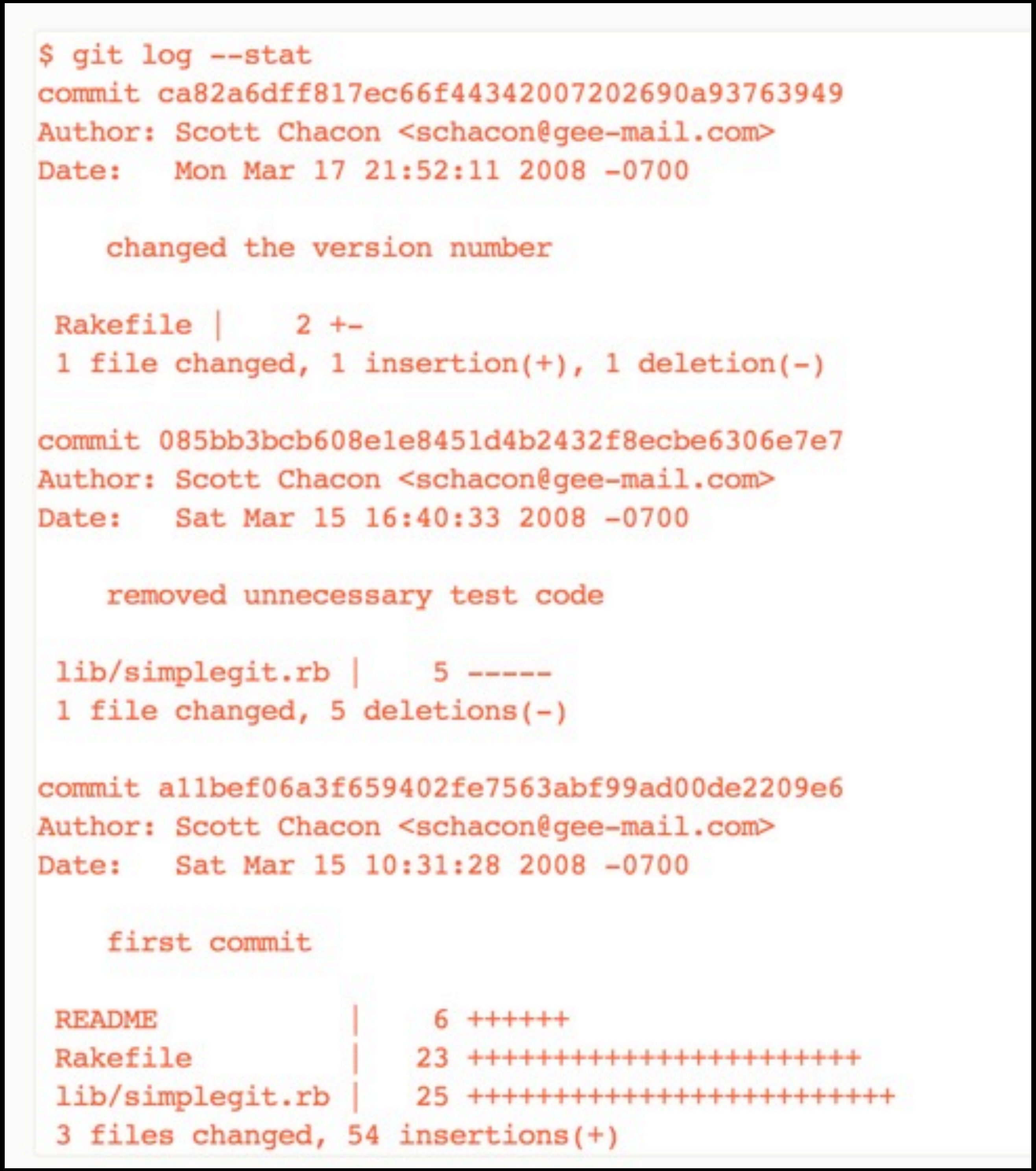
```
$ git log -p -2
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number

diff --git a/Rakefile b/Rakefile
index a874b73..8f94139 100644
--- a/Rakefile
+++ b/Rakefile
@@ -5,5 +5,5 @@ require 'rake/gempackagetask'
 spec = Gem::Specification.new do |s|
     s.name      =   "simplegit"
-    s.version   =   "0.1.0"
+    s.version   =   "0.1.1"
     s.author    =   "Scott Chacon"
     s.email     =   "schacon@gee-mail.com
```

git log --word-diff



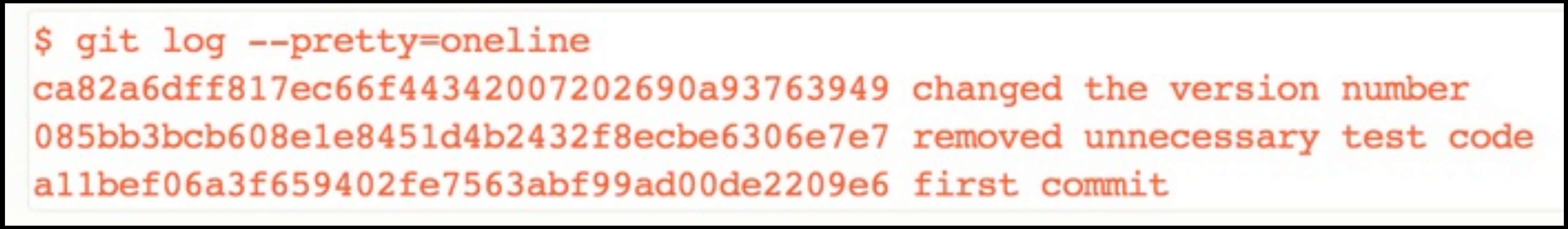
```
$ git log -U1 --word-diff
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number

diff --git a/Rakefile b/Rakefile
index a874b73..8f94139 100644
--- a/Rakefile
+++ b/Rakefile
@@ -7,3 +7,3 @@ spec = Gem::Specification.new do |s|
    s.name      =   "simplegit"
    s.version   =   [-"0.1.0"-]{+"0.1.1"+}
    s.author    =   "Scott Chacon"
```

git log --stat

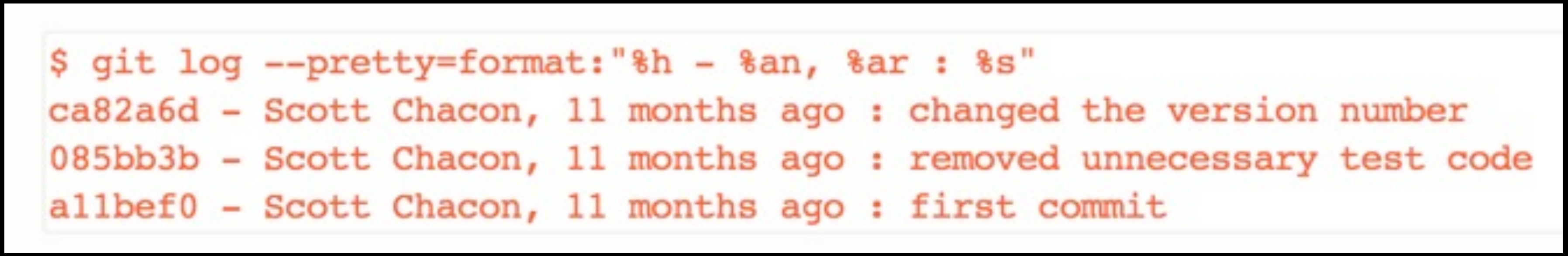


```
$ git log --stat
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number

 Rakefile |    2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

 lib/simplegit.rb |    5 -----
 1 file changed, 5 deletions(-)

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 10:31:28 2008 -0700

    first commit

 README           |    6 ++++++
 Rakefile         |   23 +++++++++++++++++++++++
 lib/simplegit.rb |   25 +++++++++++++++++++++++++
 3 files changed, 54 insertions(+)
```

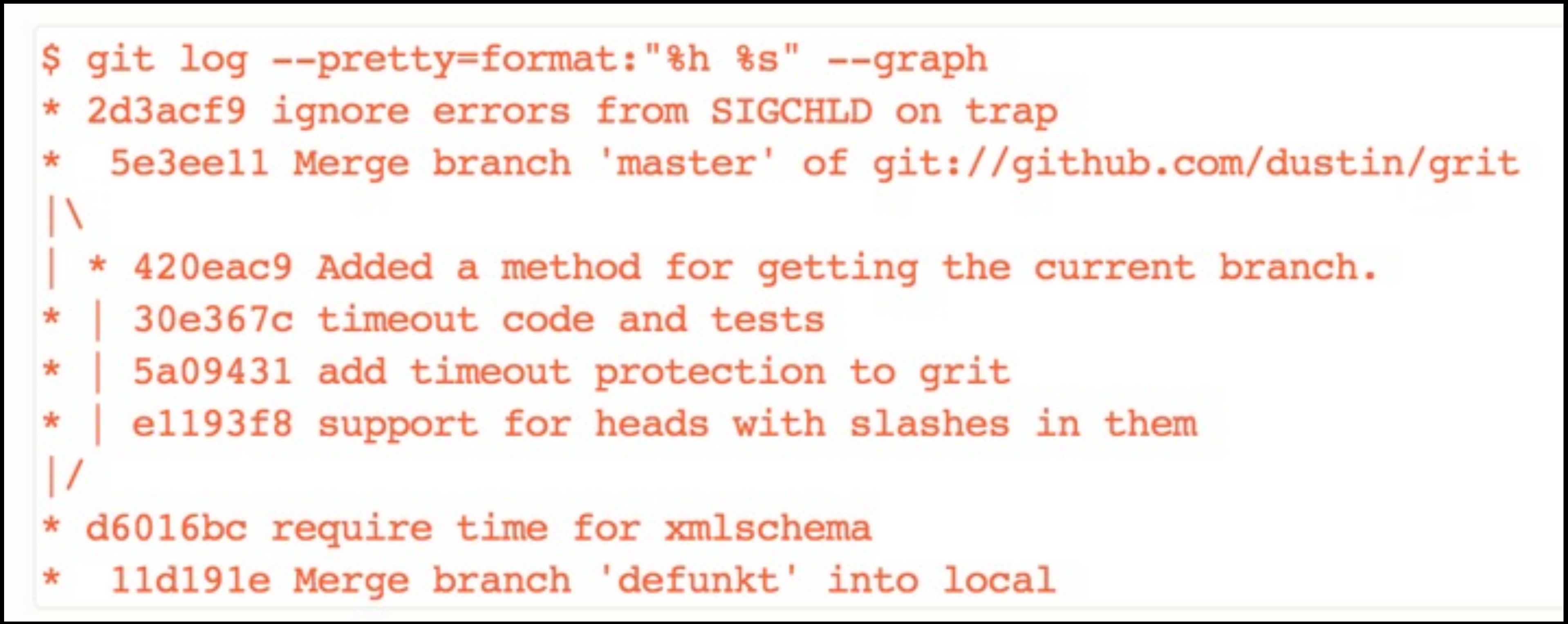
git log hacking



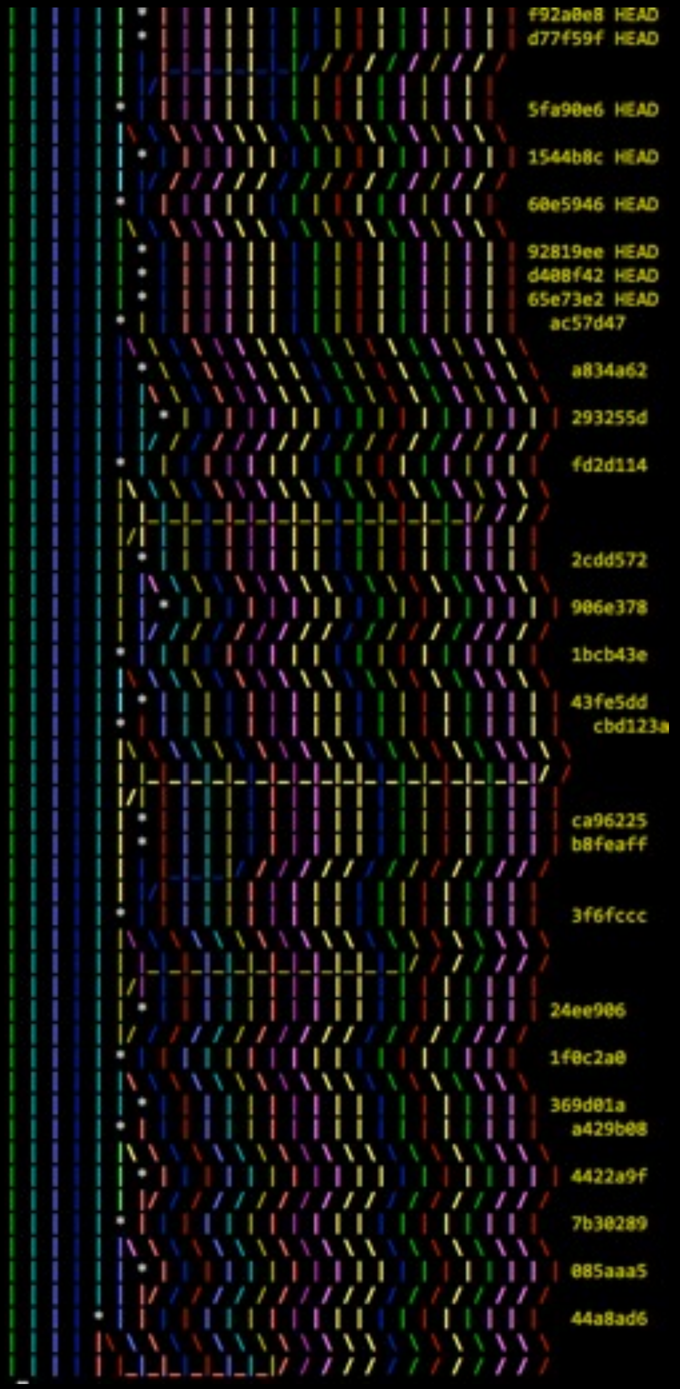
```
$ git log --pretty=oneline
ca82a6dff817ec66f44342007202690a93763949 changed the version number
085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7 removed unnecessary test code
a11bef06a3f659402fe7563abf99ad00de2209e6 first commit
```



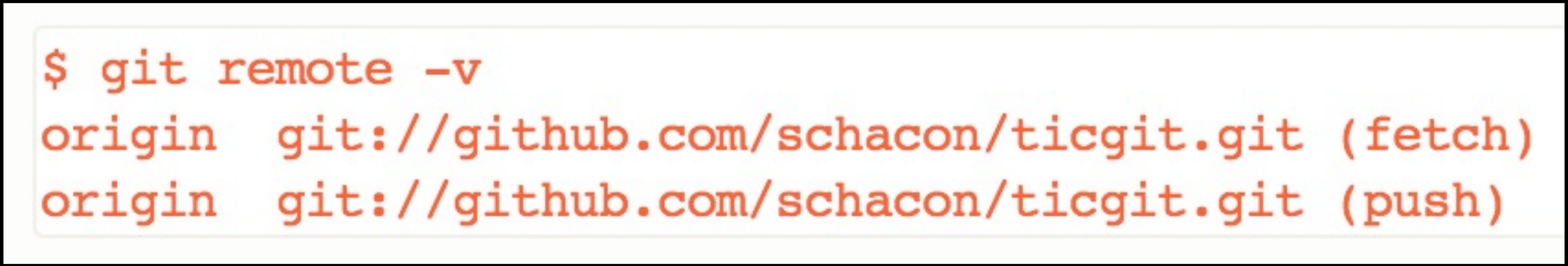
```
$ git log --pretty=format:"%h - %an, %ar : %s"
ca82a6d - Scott Chacon, 11 months ago : changed the version number
085bb3b - Scott Chacon, 11 months ago : removed unnecessary test code
a11bef0 - Scott Chacon, 11 months ago : first commit
```



```
$ git log --pretty=format:"%h %s" --graph
* 2d3acf9 ignore errors from SIGCHLD on trap
*  5e3ee11 Merge branch 'master' of git://github.com/dustin/grit
|\
| * 420eac9 Added a method for getting the current branch.
* | 30e367c timeout code and tests
* | 5a09431 add timeout protection to grit
* | e1193f8 support for heads with slashes in them
|/
* d6016bc require time for xmlschema
*  11d191e Merge branch 'defunkt' into local
```

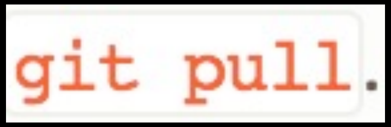


git remote

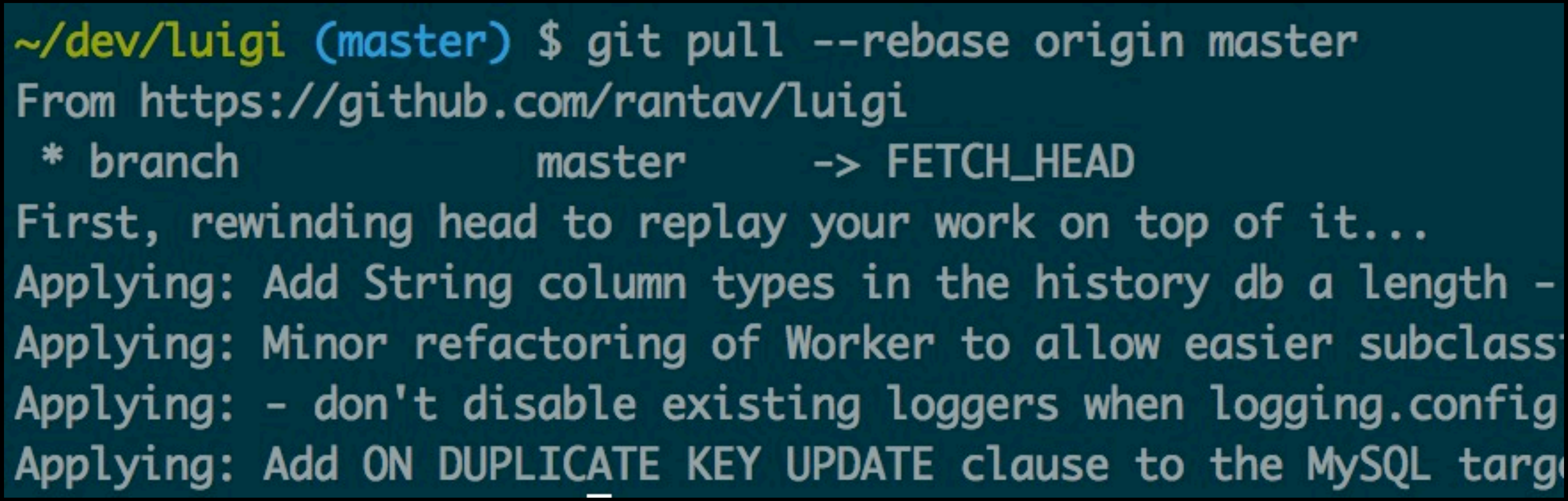


```
$ git remote -v
origin  git://github.com/schacon/ticgit.git (fetch)
origin  git://github.com/schacon/ticgit.git (push)
```

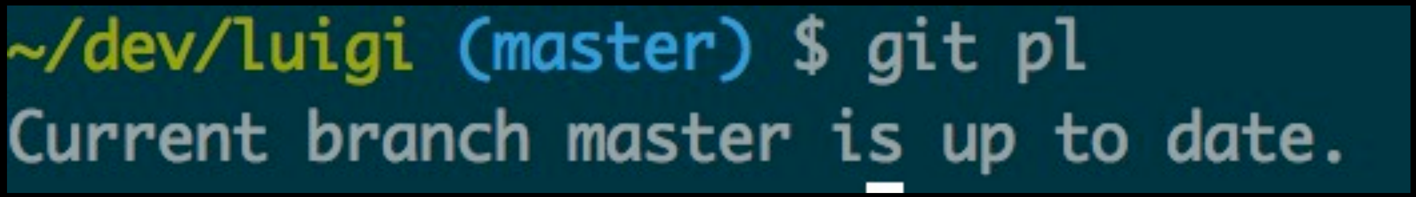
git pull



`git pull`.



```
~/dev/luigi (master) $ git pull --rebase origin master
From https://github.com/rantav/luigi
 * branch            master     -> FETCH_HEAD
First, rewinding head to replay your work on top of it...
Applying: Add String column types in the history db a length -
Applying: Minor refactoring of Worker to allow easier subclass
Applying: - don't disable existing loggers when logging.config
Applying: Add ON DUPLICATE KEY UPDATE clause to the MySQL targ
```

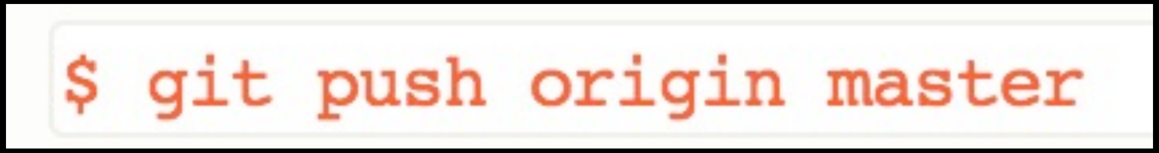


```
~/dev/luigi (master) $ git pl
Current branch master is up to date.
```

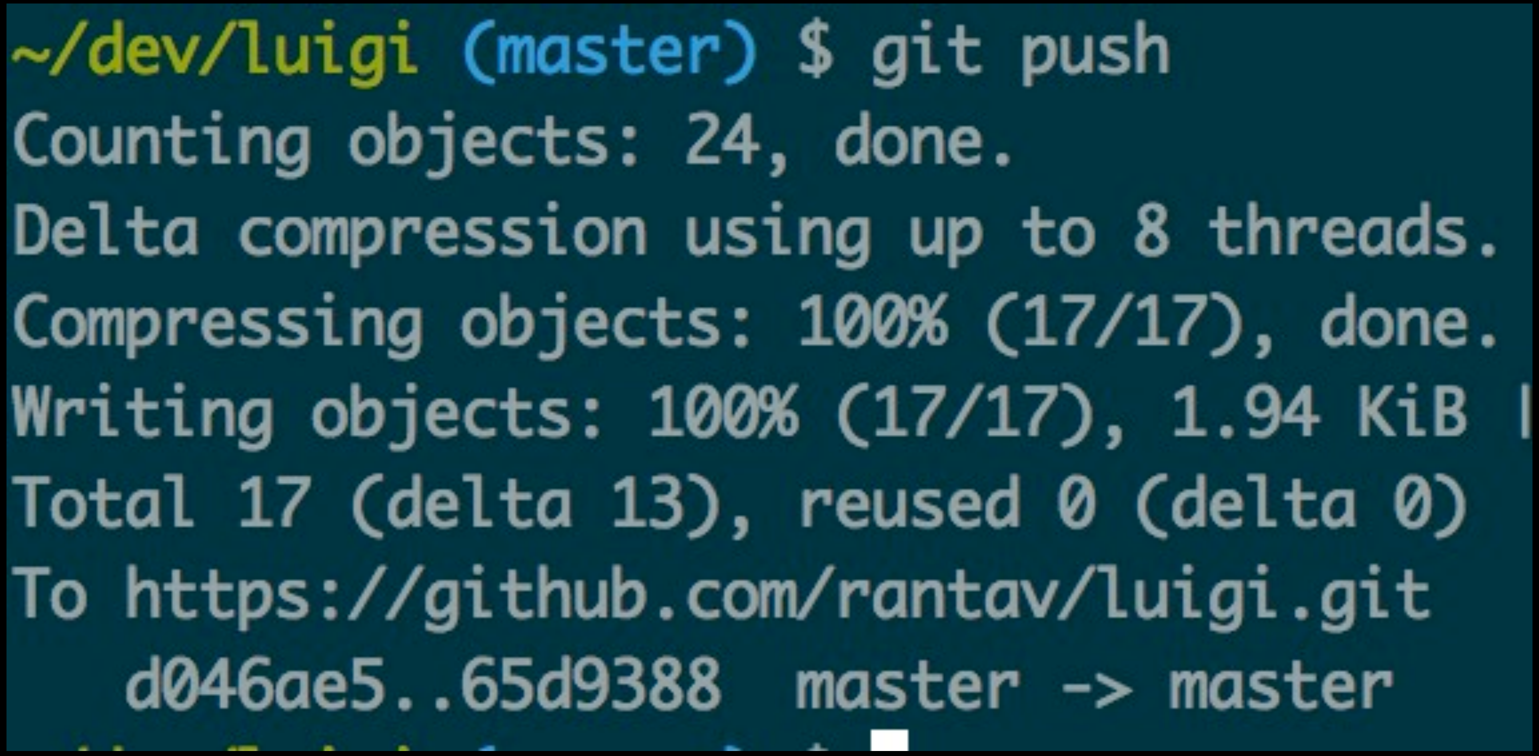
rebase or no?

- Prefer rebase
  - this keeps the commit log cleaner
  - **`git pl`**
- Sometimes an actual merge is needed
  - So don’t rebase
  - For example, when merging branches

git push

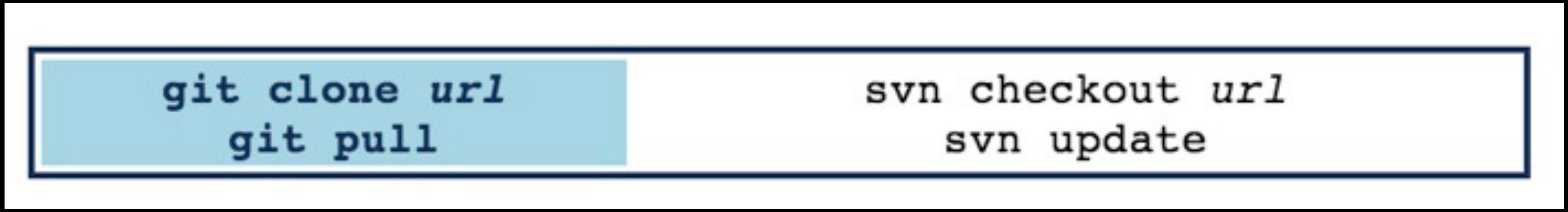


```
$ git push origin master
```

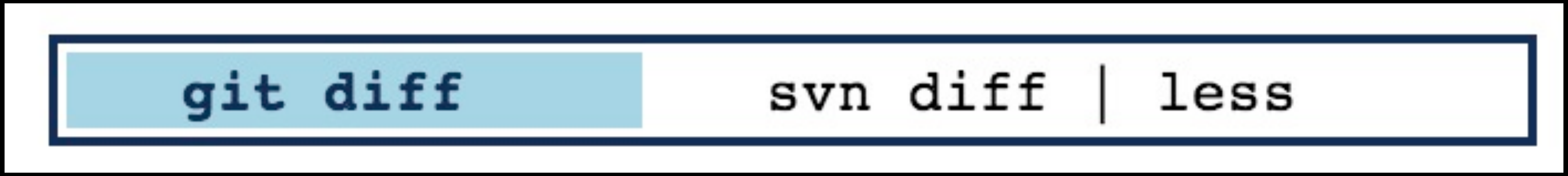


```
~/dev/luigi (master) $ git push
Counting objects: 24, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (17/17), done.
Writing objects: 100% (17/17), 1.94 KiB
Total 17 (delta 13), reused 0 (delta 0)
To https://github.com/rantav/luigi.git
   d046ae5..65d9388  master -> master
```

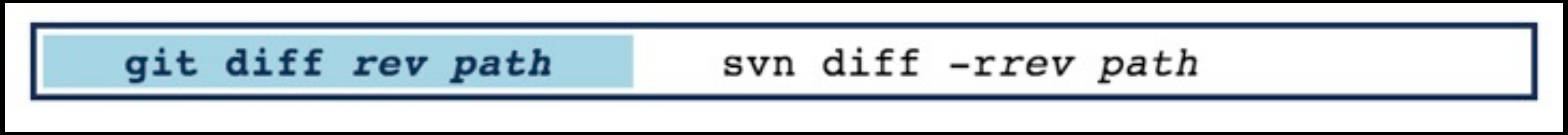
Git for SVN



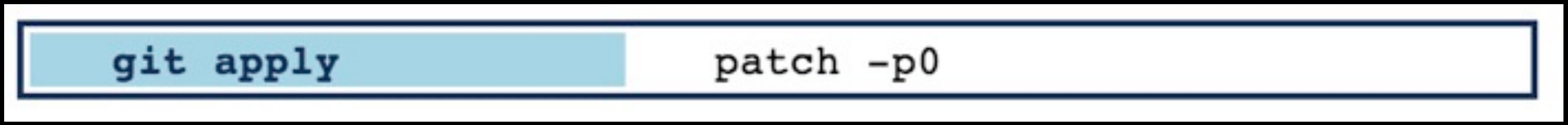
`git clone` *`url`*
`git pull`
	`svn checkout` *`url`*
	`svn update`



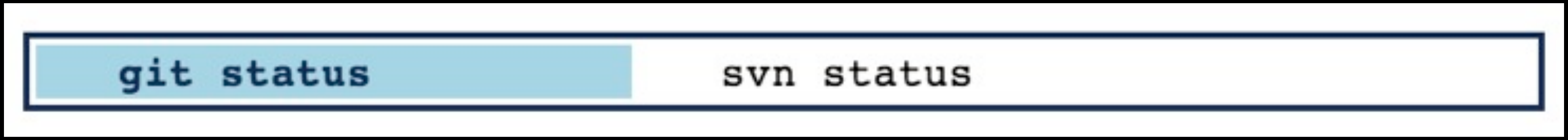
`git diff`	`svn diff | less`



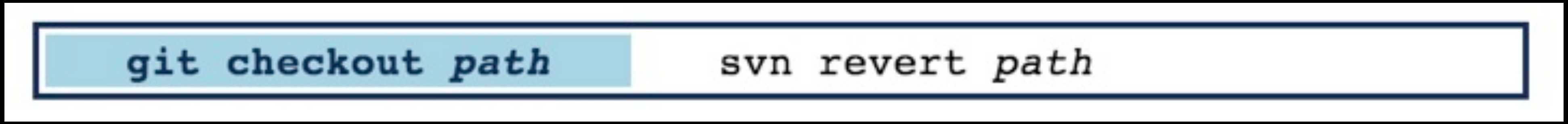
`git diff` *`rev path`*	`svn diff -r`*`rev path`*



`git apply`	`patch -p0`

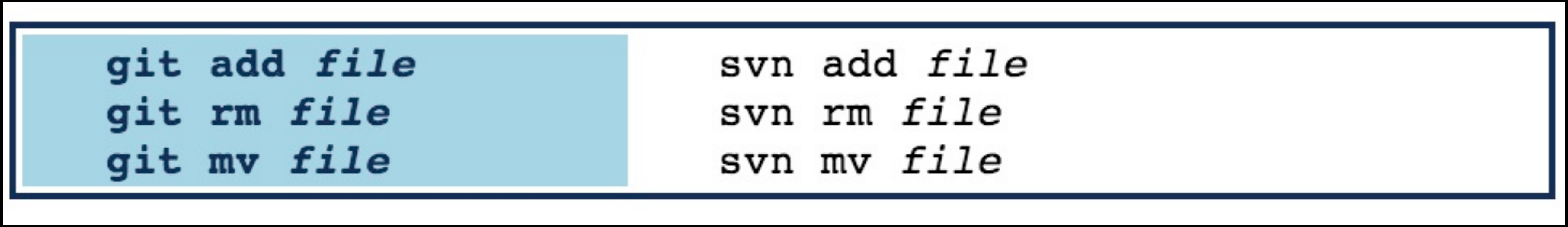


`git status`	`svn status`

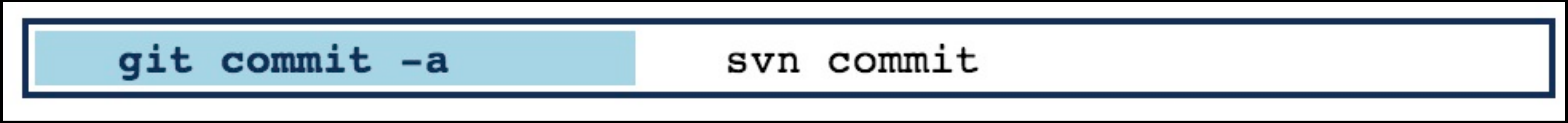


`git checkout` *`path`*	`svn revert` *`path`*

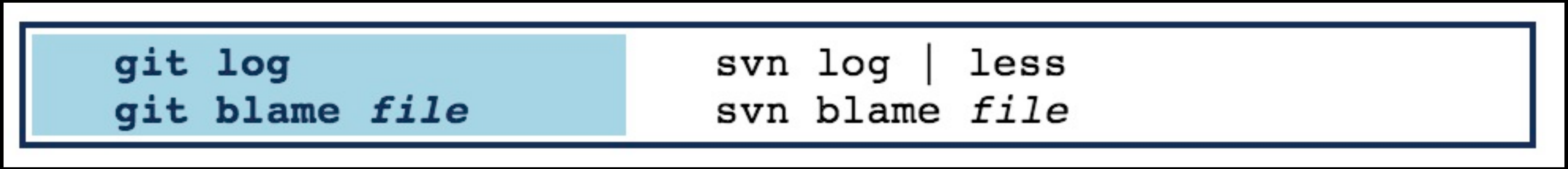
Git for SVN



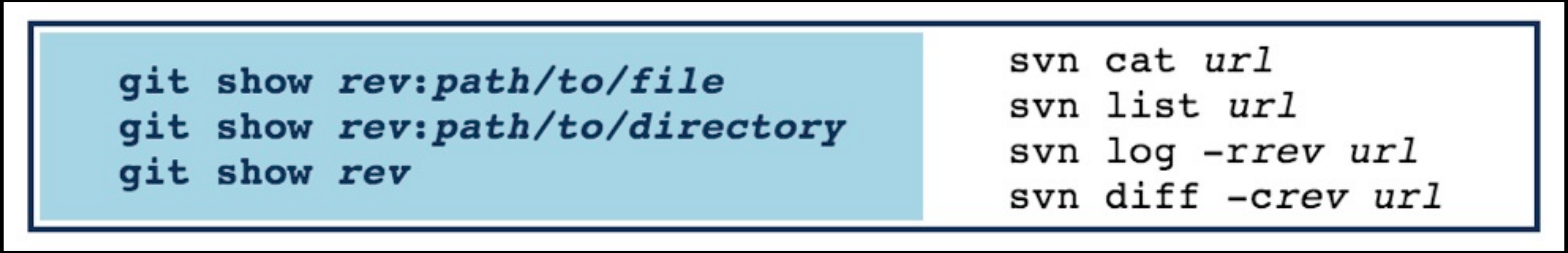
git add *file*	svn add *file*
git rm *file*	svn rm *file*
git mv *file*	svn mv *file*



git commit -a	svn commit

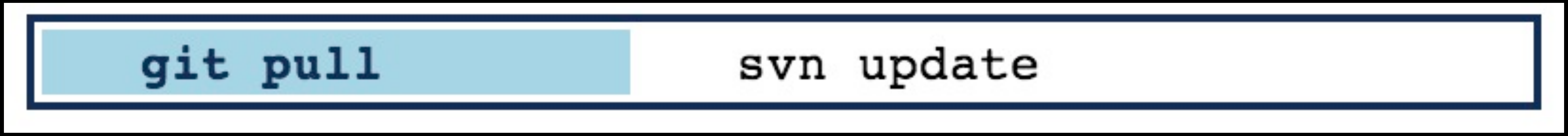


git log	svn log | less
git blame *file*	svn blame *file*

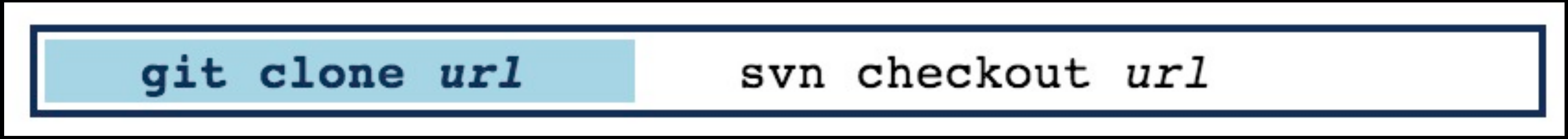


git show *rev:path/to/file*	svn cat *url*
git show *rev:path/to/directory*	svn list *url*
git show *rev*	svn log -r*rev url*
	svn diff -c*rev url*

Git for SVN

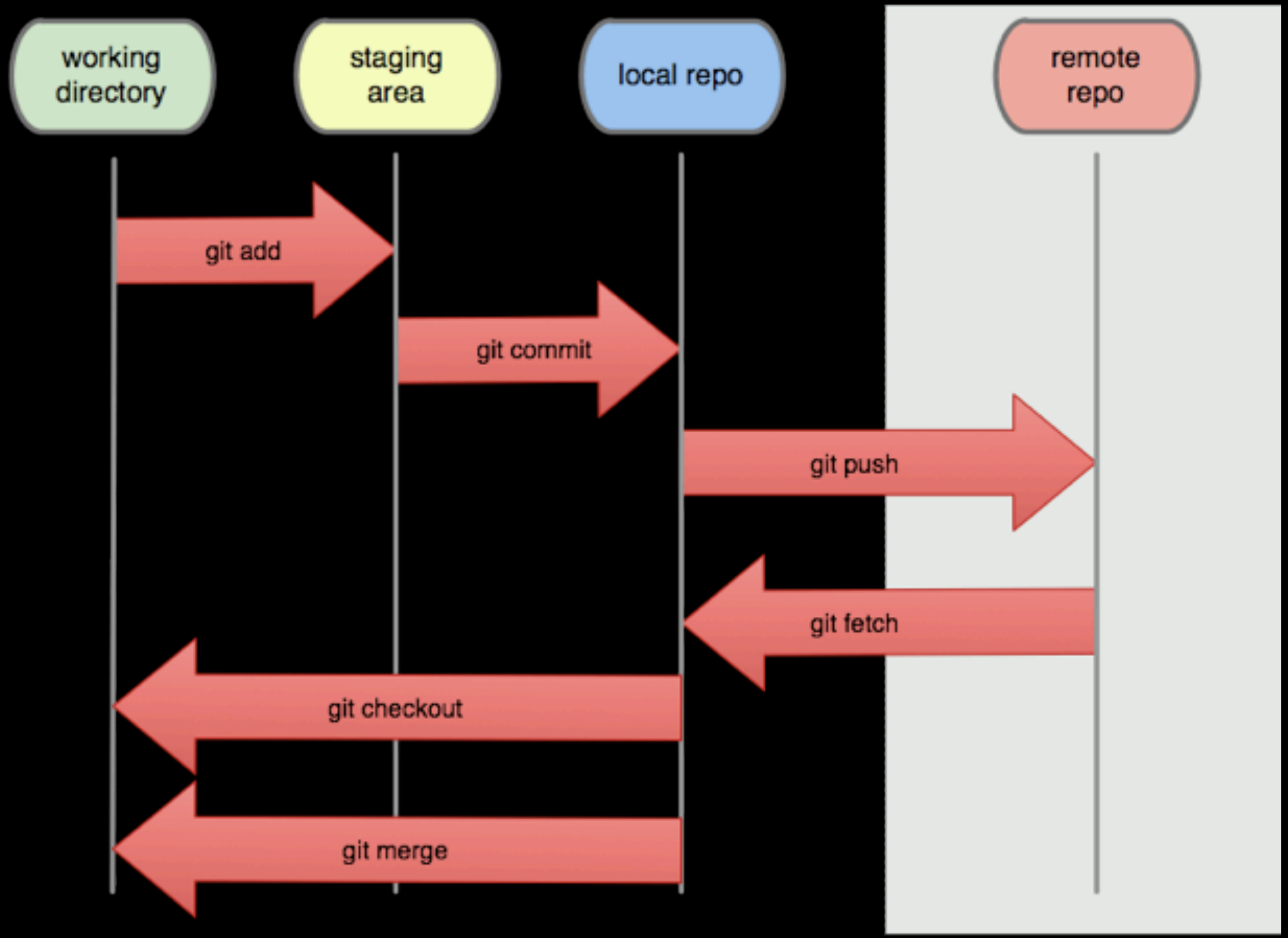


`git pull`	`svn update`

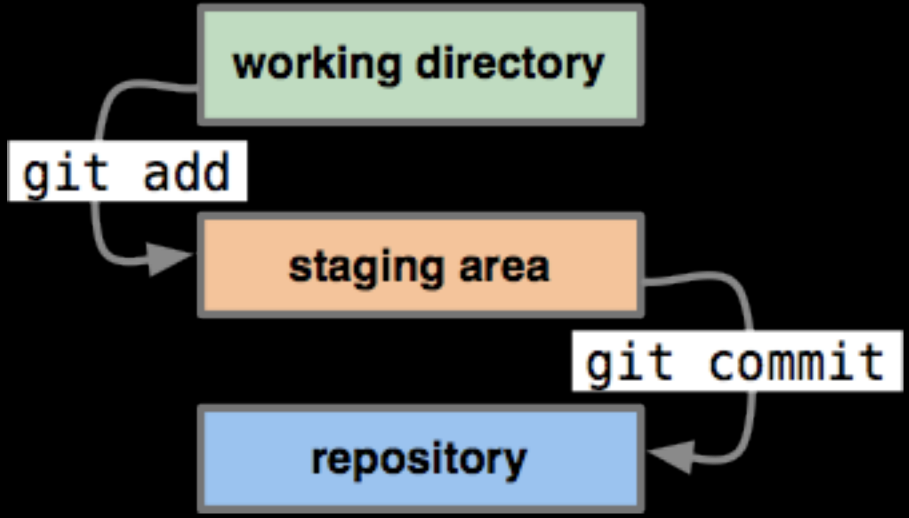


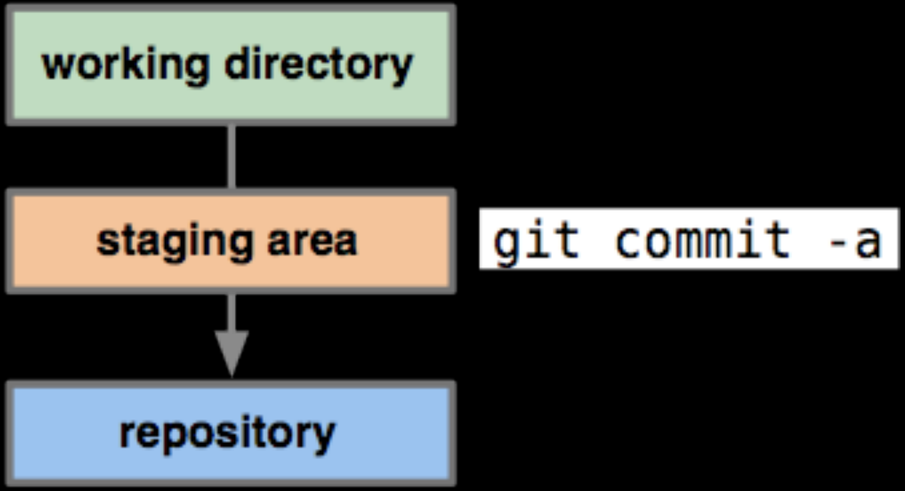
`git clone` *`url`*	`svn checkout` *`url`*

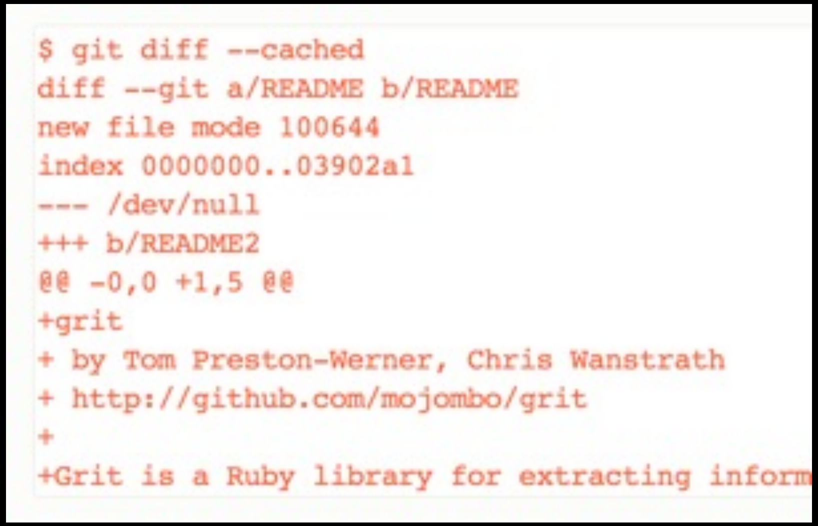
Git Lifecycle



The Staging Area
(AKA The Index)

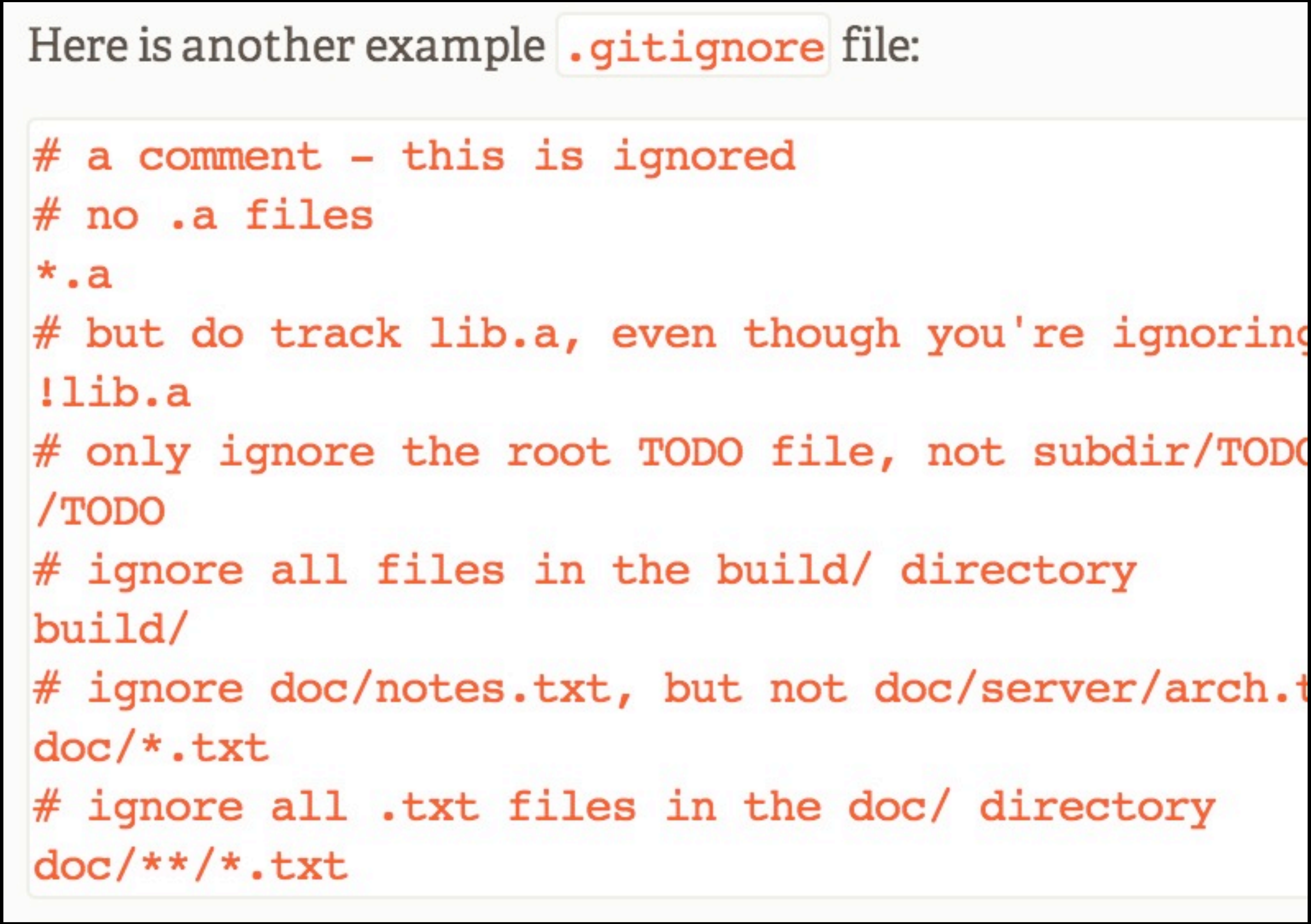






```
$ git diff --cached
diff --git a/README b/README
new file mode 100644
index 0000000..03902a1
--- /dev/null
+++ b/README2
@@ -0,0 +1,5 @@
+grit
+ by Tom Preston-Werner, Chris Wanstrath
+ http://github.com/mojombo/grit
+
+Grit is a Ruby library for extracting inform
```

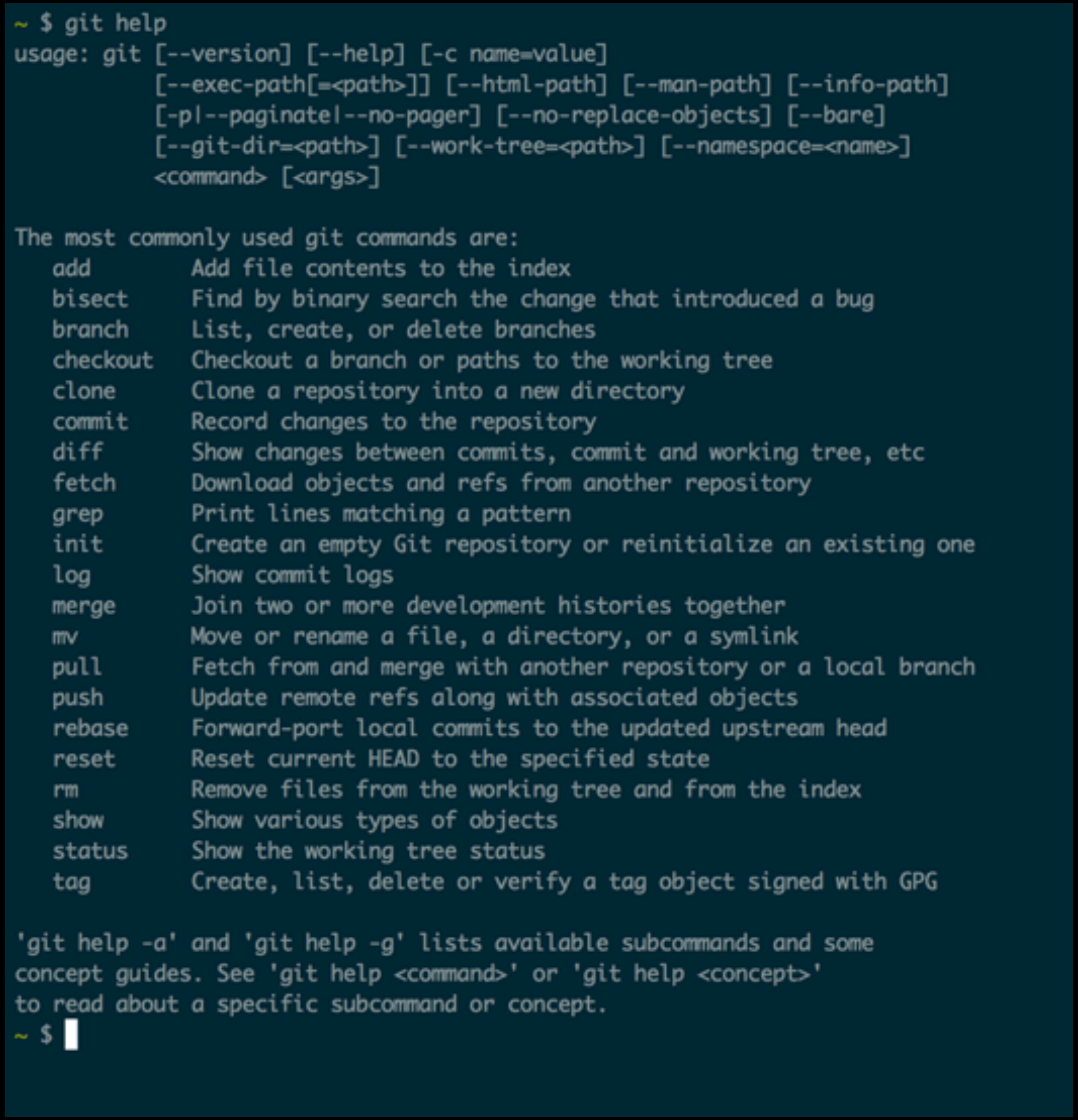
.gitignore



Here is another example `.gitignore` file:

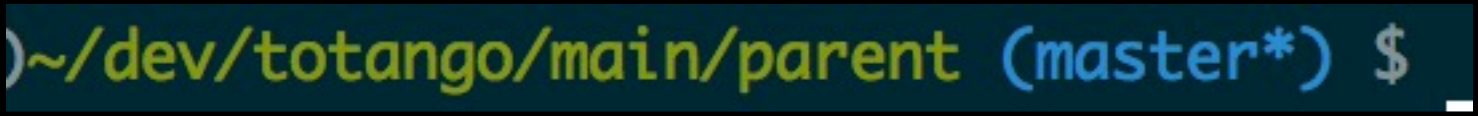
```
# a comment - this is ignored
# no .a files
*.a
# but do track lib.a, even though you're ignorin
!lib.a
# only ignore the root TODO file, not subdir/TOD
/TODO
# ignore all files in the build/ directory
build/
# ignore doc/notes.txt, but not doc/server/arch.
doc/*.txt
# ignore all .txt files in the doc/ directory
doc/**/*.txt
```

Learning More

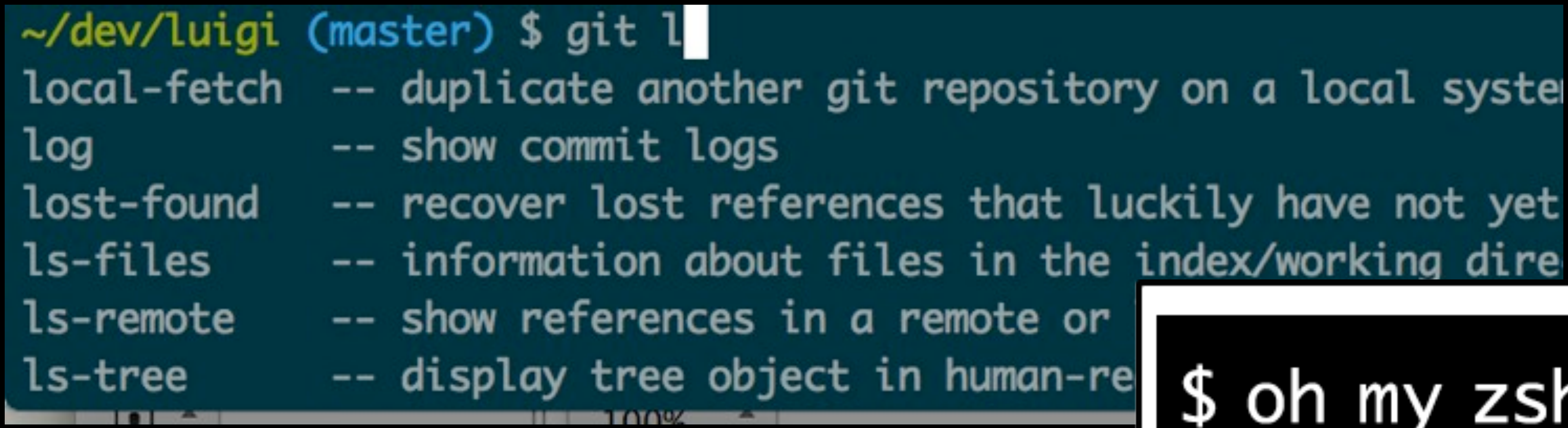


```
~ $ git help
usage: git [--version] [--help] [-c name=value]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<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
   init       Create an empty Git repository or reinitialize an existing one
   log        Show commit logs
   merge      Join two or more development histories together
   mv         Move or rename a file, a directory, or a symlink
   pull       Fetch from and merge with another repository or a local branch
   push       Update remote refs along with associated objects
   rebase     Forward-port local commits to the updated upstream head
   reset      Reset current HEAD to the specified state
   rm         Remove files from the working tree and from the index
   show       Show various types of objects
   status     Show the working tree status
   tag        Create, list, delete or verify a tag object signed with GPG

'git help -a' and 'git help -g' lists available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
~ $
```

Bonus: zsh integration

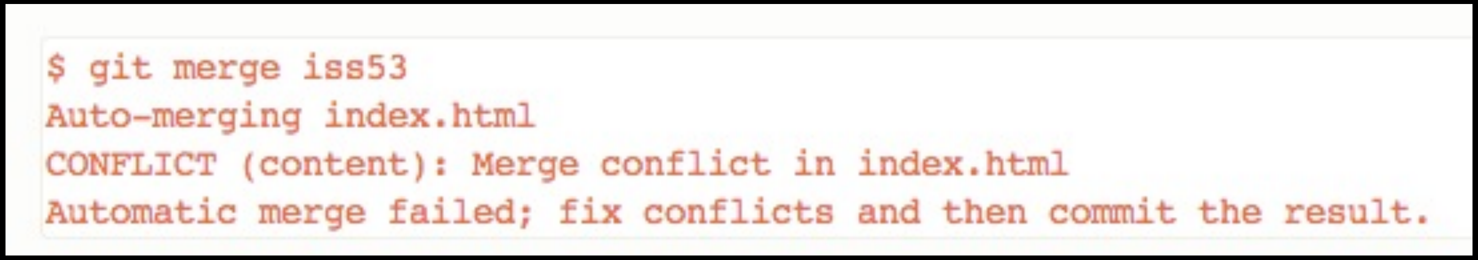


```
)~/dev/totango/main/parent (master*) $ _
```

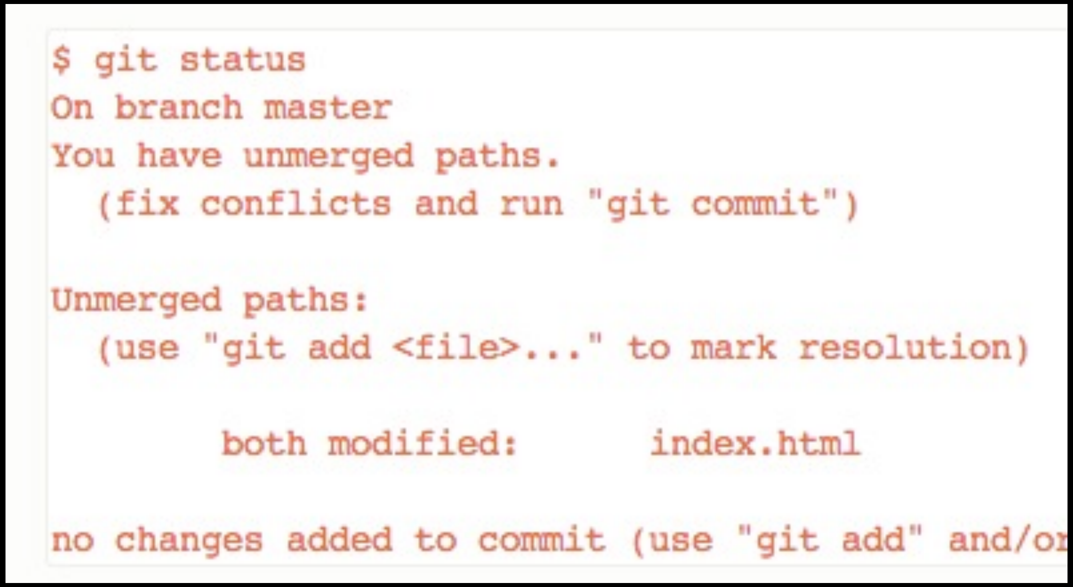


```
~/dev/luigi (master) $ git l
local-fetch   -- duplicate another git repository on a local syste
log           -- show commit logs
lost-found    -- recover lost references that luckily have not yet
ls-files      -- information about files in the index/working dire
ls-remote     -- show references in a remote or
ls-tree       -- display tree object in human-re
```

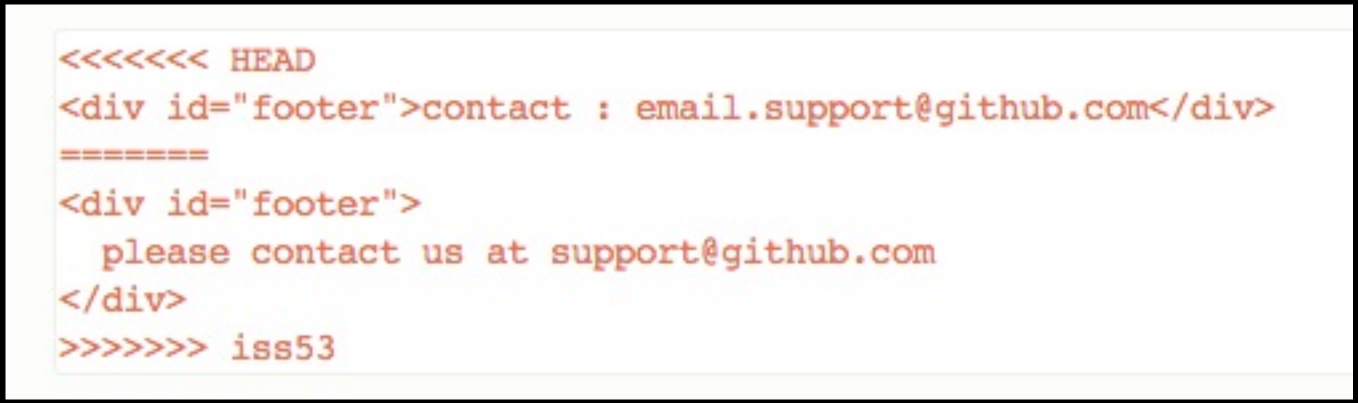
Extra: Merge



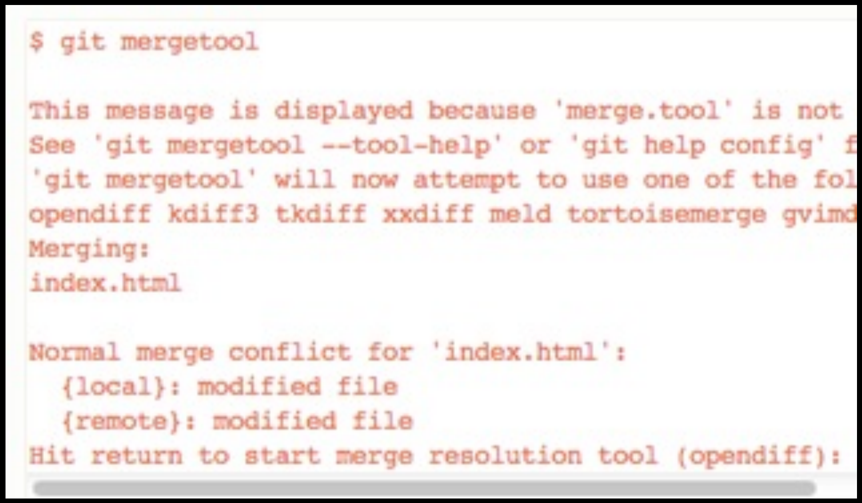
```
$ git merge iss53
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```



```
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")

Unmerged paths:
  (use "git add <file>..." to mark resolution)

        both modified:      index.html

no changes added to commit (use "git add" and/o
```

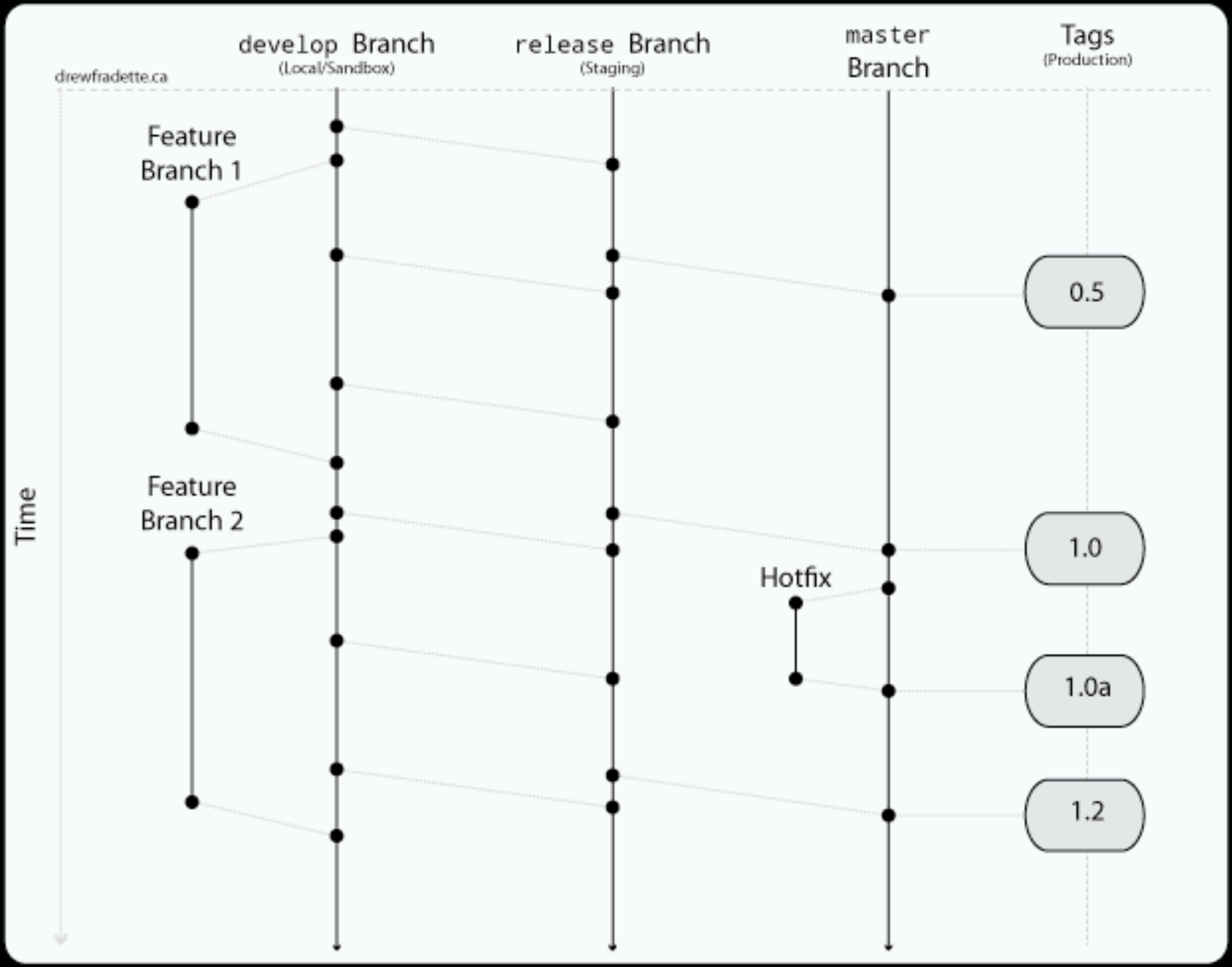


```
<<<<<<< HEAD
<div id="footer">contact : email.support@github.com</div>
=======
<div id="footer">
  please contact us at support@github.com
</div>
>>>>>>> iss53
```



```
$ git mergetool

This message is displayed because 'merge.tool' is not
See 'git mergetool --tool-help' or 'git help config' f
'git mergetool' will now attempt to use one of the fol
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimd
Merging:
index.html

Normal merge conflict for 'index.html':
  {local}: modified file
  {remote}: modified file
Hit return to start merge resolution tool (opendiff):
```

Not Today: Branching



Totango Links

- Wiki
- Repo
- CI

References

AKA where did I steal from?

http://git-scm.com/about

http://thkoch2001.github.io/whygitisbetter/

http://www.slideshare.net/manishchaks/subversion-to-git-migration

http://www.git-tower.com/blog/git-for-subversion-users-cheat-sheet-detail/

http://git.or.cz/course/svn.html