Everyday Git

Inside Out


git

Git is a state of mind.

Git is an incomprehensible mess.

Git is hard.

Git is a way of life.

Git is too terrible for mere mortals to understand.

Git is a state of mind.

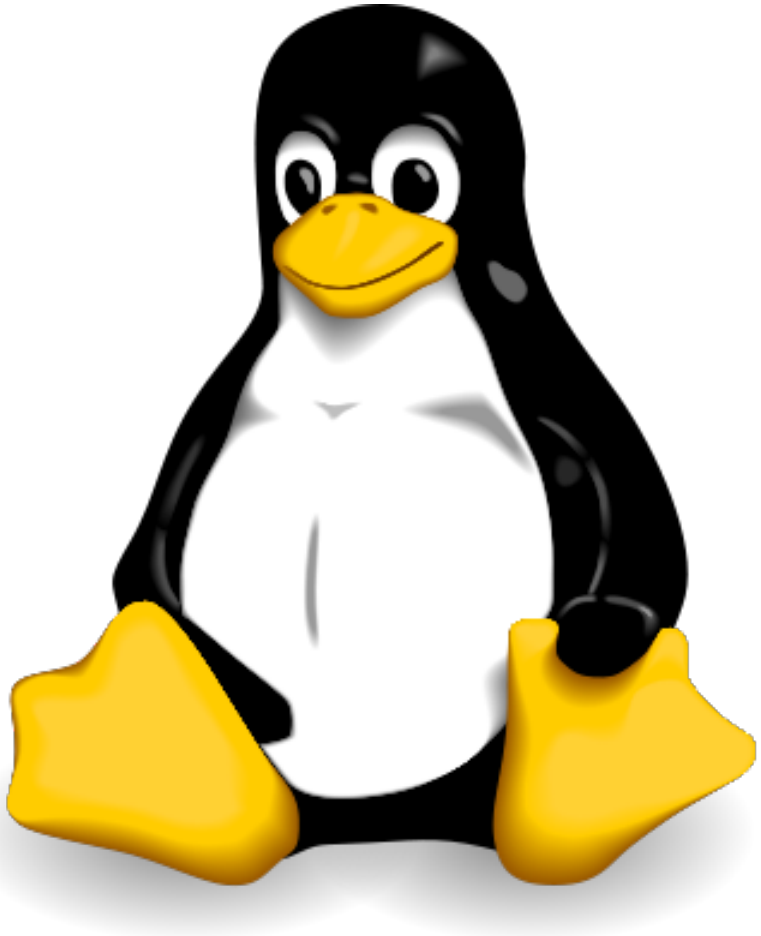
Git is an incomprehensible mess.

Git is ~~hard~~ different.

Git is a way of life.

Git is too terrible for mere mortals to understand.

C:\>




BitKeeper

https://www.youtube.com/watch?v=4XpnKHJAok8

mercurial

Bazaar

Where to get help

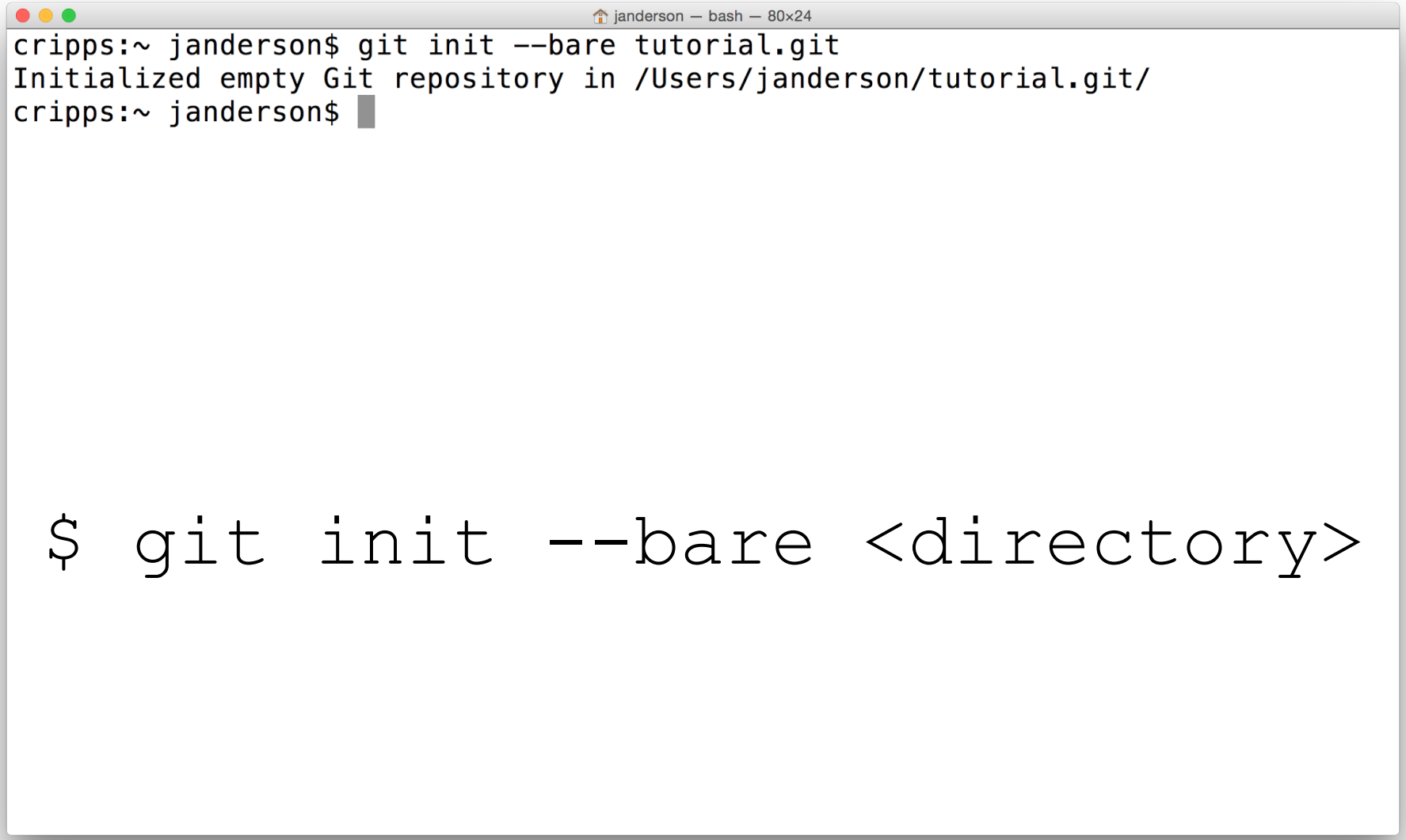
- Pro Git

  http://git-scm.com/book/en/v2/
- Git Reference

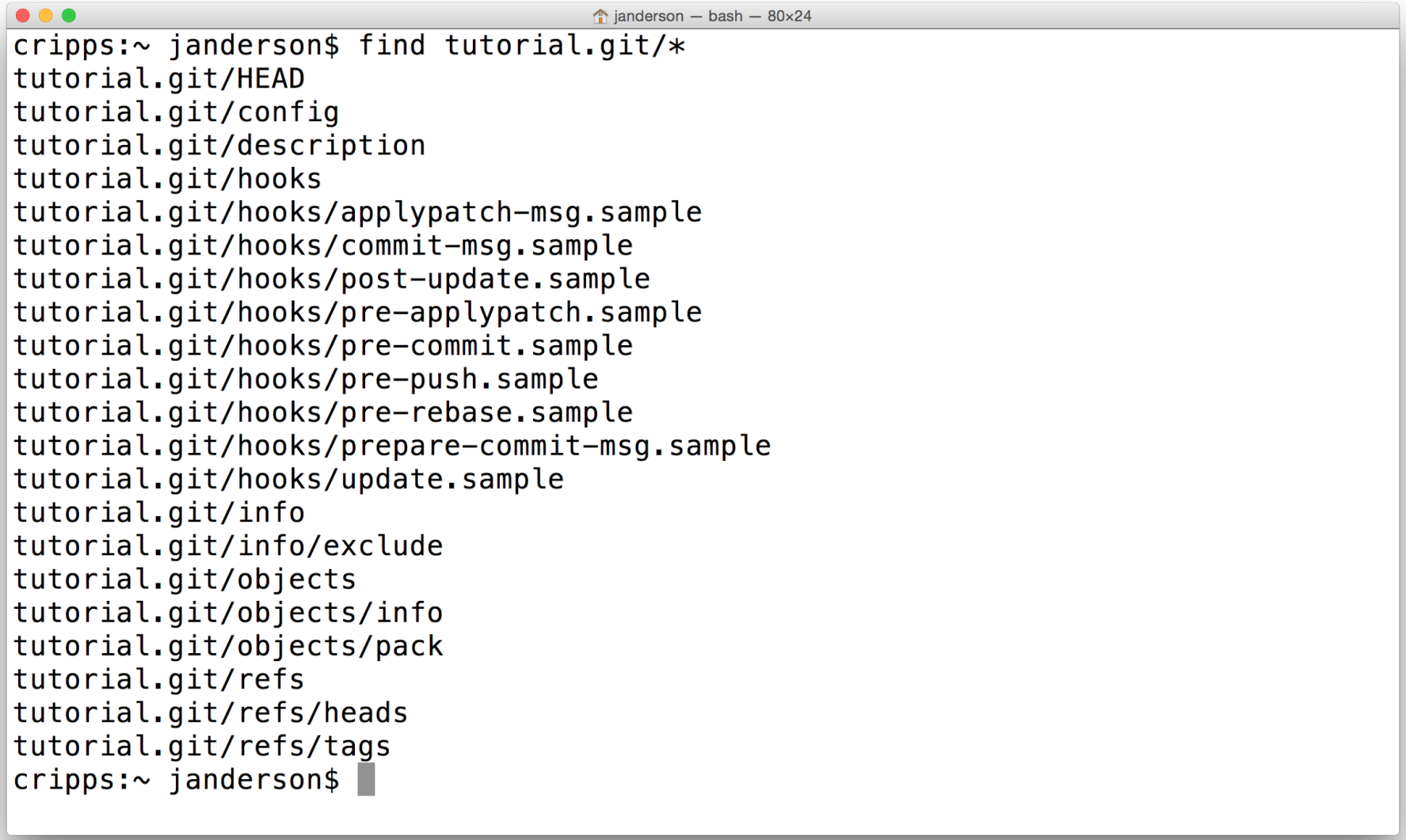
  http://gitref.org
- Git Community Book

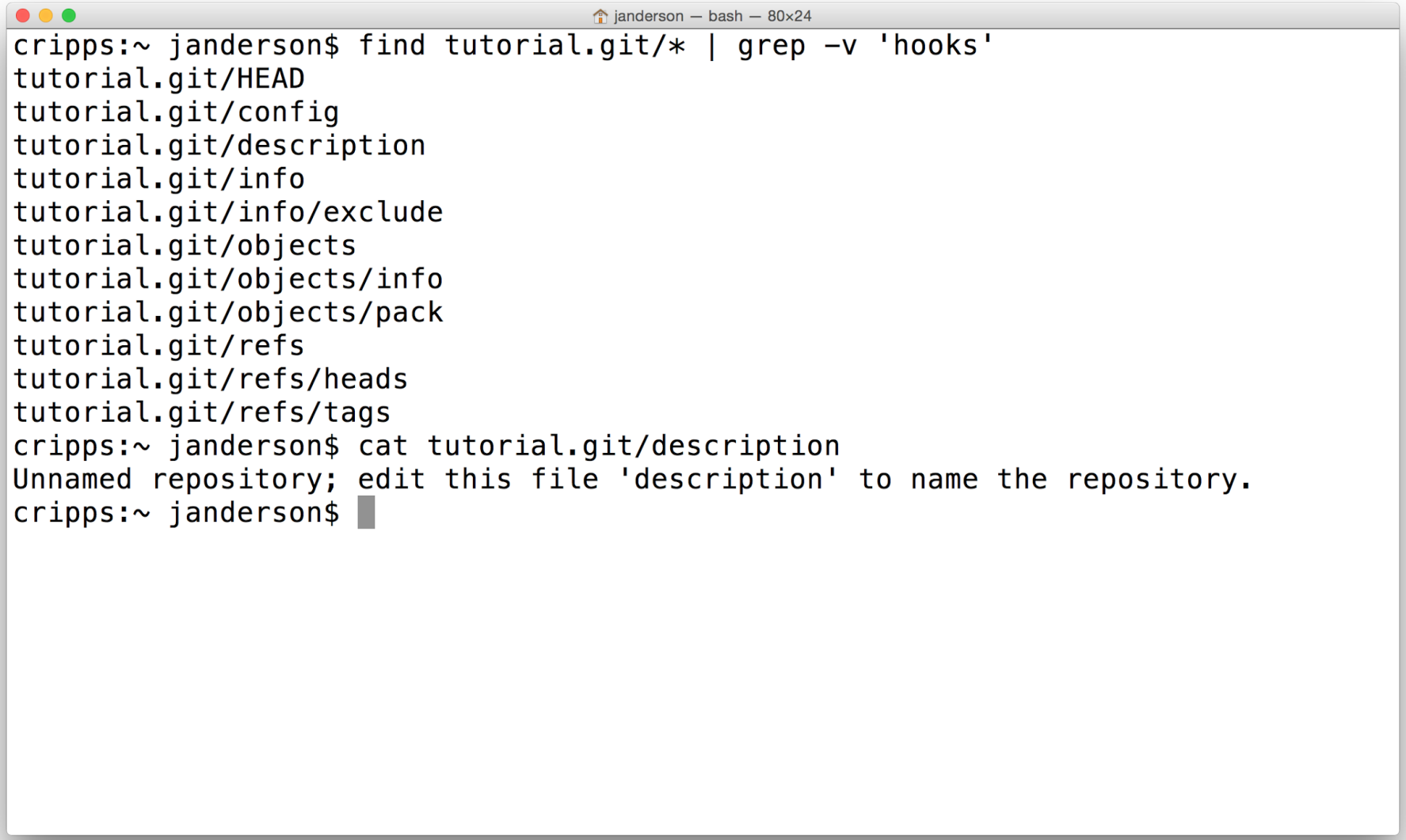
  http://alx.github.io/gitbook/
- `$ git help <command>`

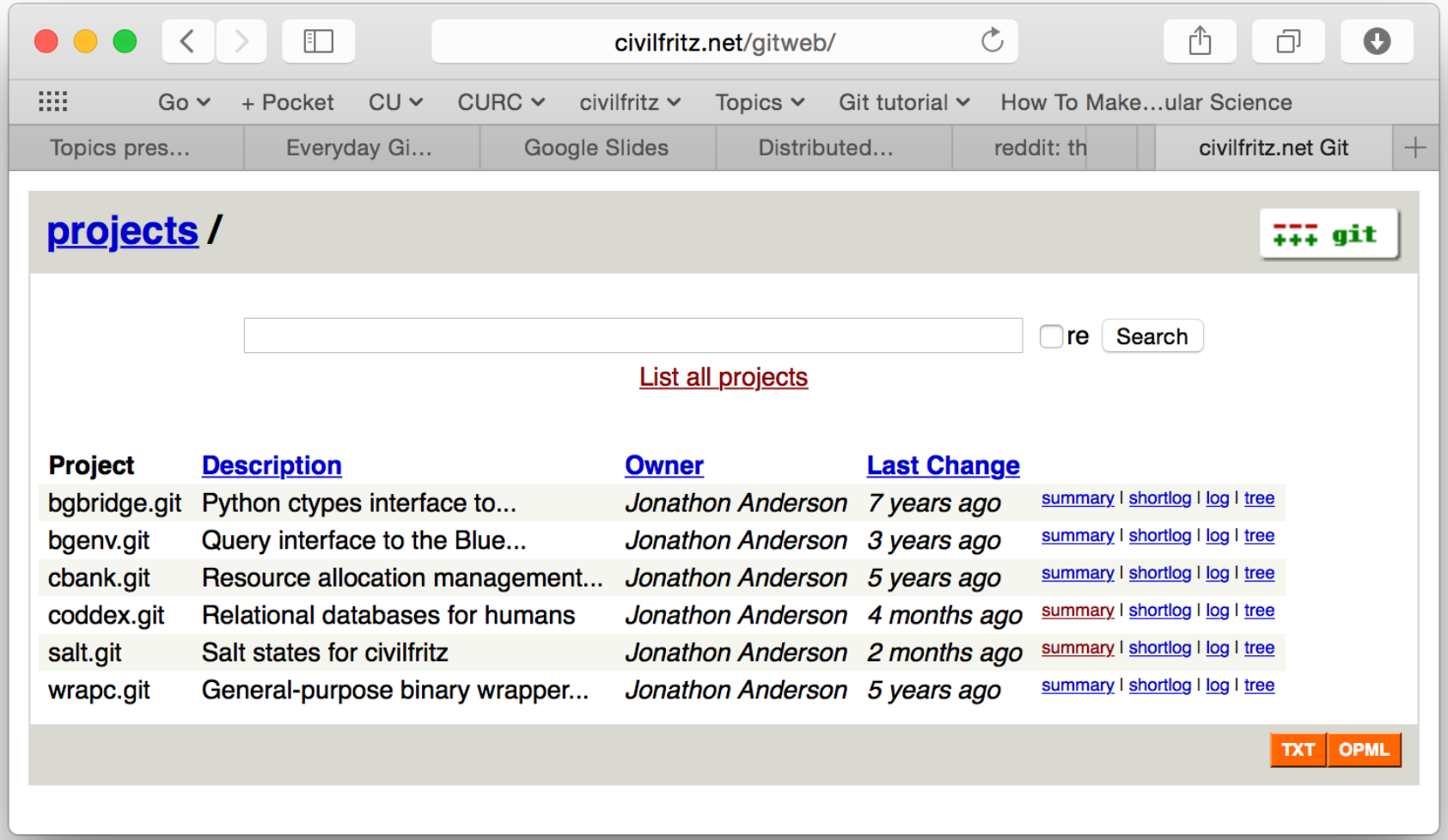
Initializing a repository

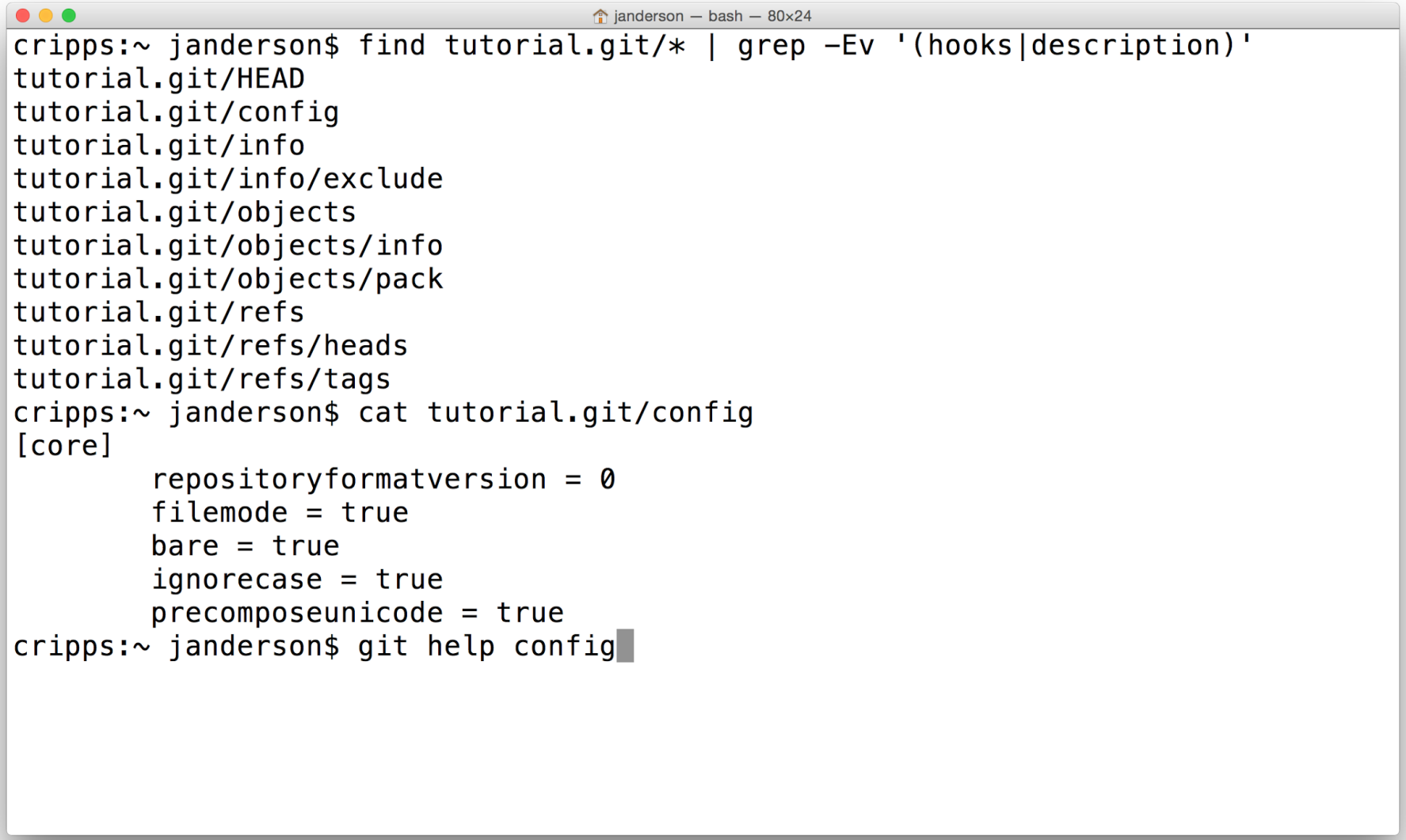


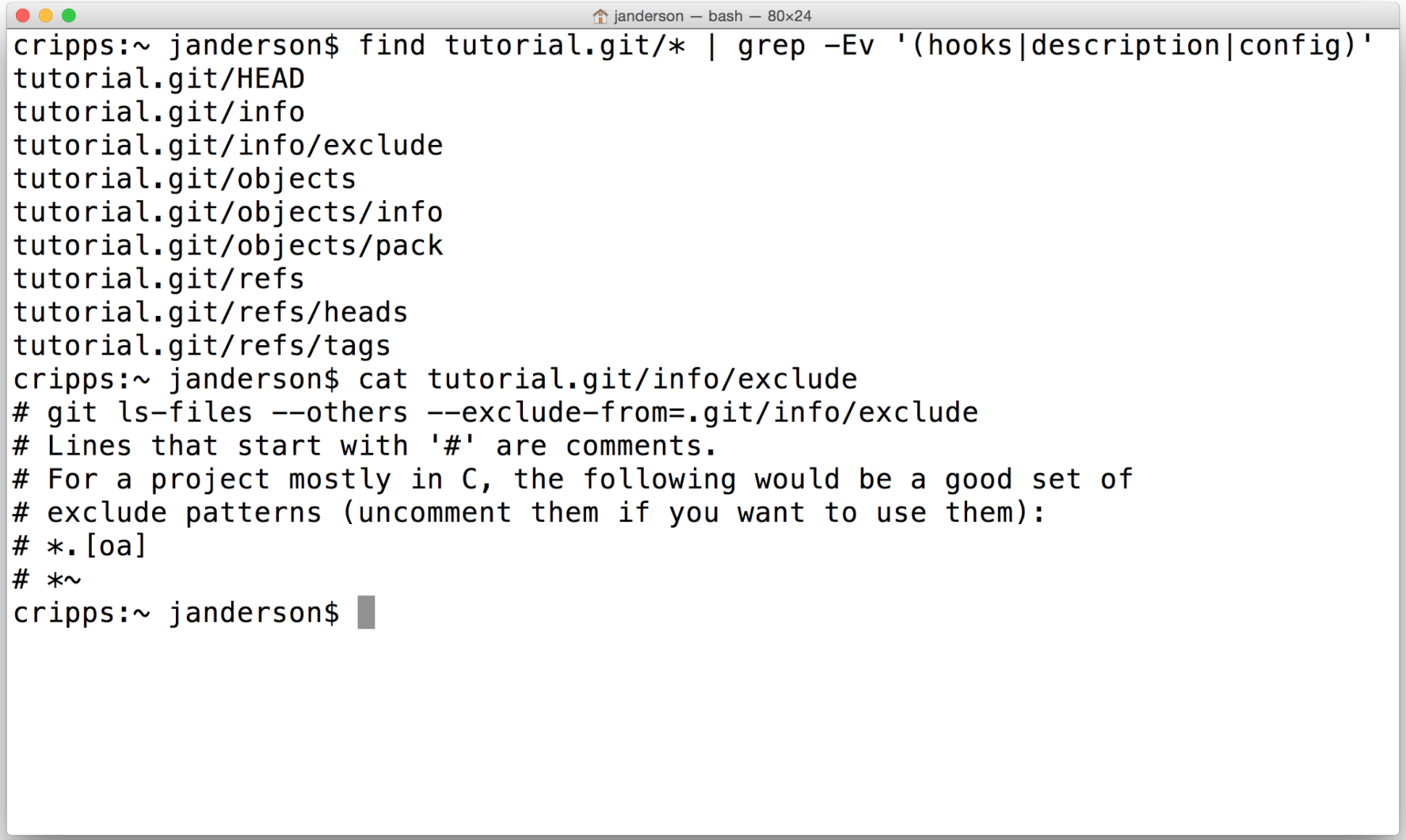
$ git init --bare <directory>

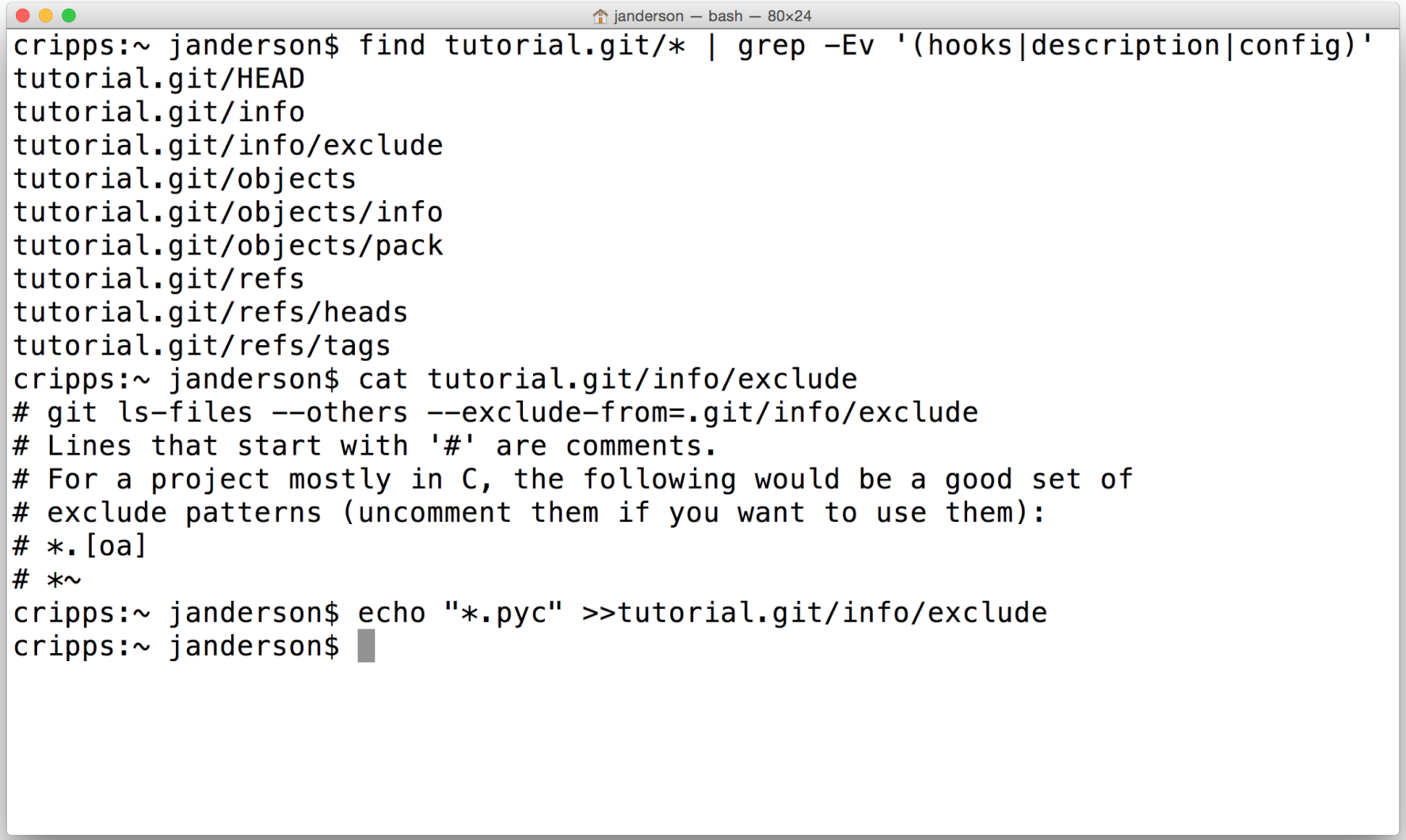

janderson — bash — 80×24
cripps:~ janderson$ find tutorial.git/*
tutorial.git/HEAD
tutorial.git/config
tutorial.git/description
tutorial.git/hooks
tutorial.git/hooks/applypatch-msg.sample
tutorial.git/hooks/commit-msg.sample
tutorial.git/hooks/post-update.sample
tutorial.git/hooks/pre-applypatch.sample
tutorial.git/hooks/pre-commit.sample
tutorial.git/hooks/pre-push.sample
tutorial.git/hooks/pre-rebase.sample
tutorial.git/hooks/prepare-commit-msg.sample
tutorial.git/hooks/update.sample
tutorial.git/info
tutorial.git/info/exclude
tutorial.git/objects
tutorial.git/objects/info
tutorial.git/objects/pack
tutorial.git/refs
tutorial.git/refs/heads
tutorial.git/refs/tags
cripps:~ janderson$

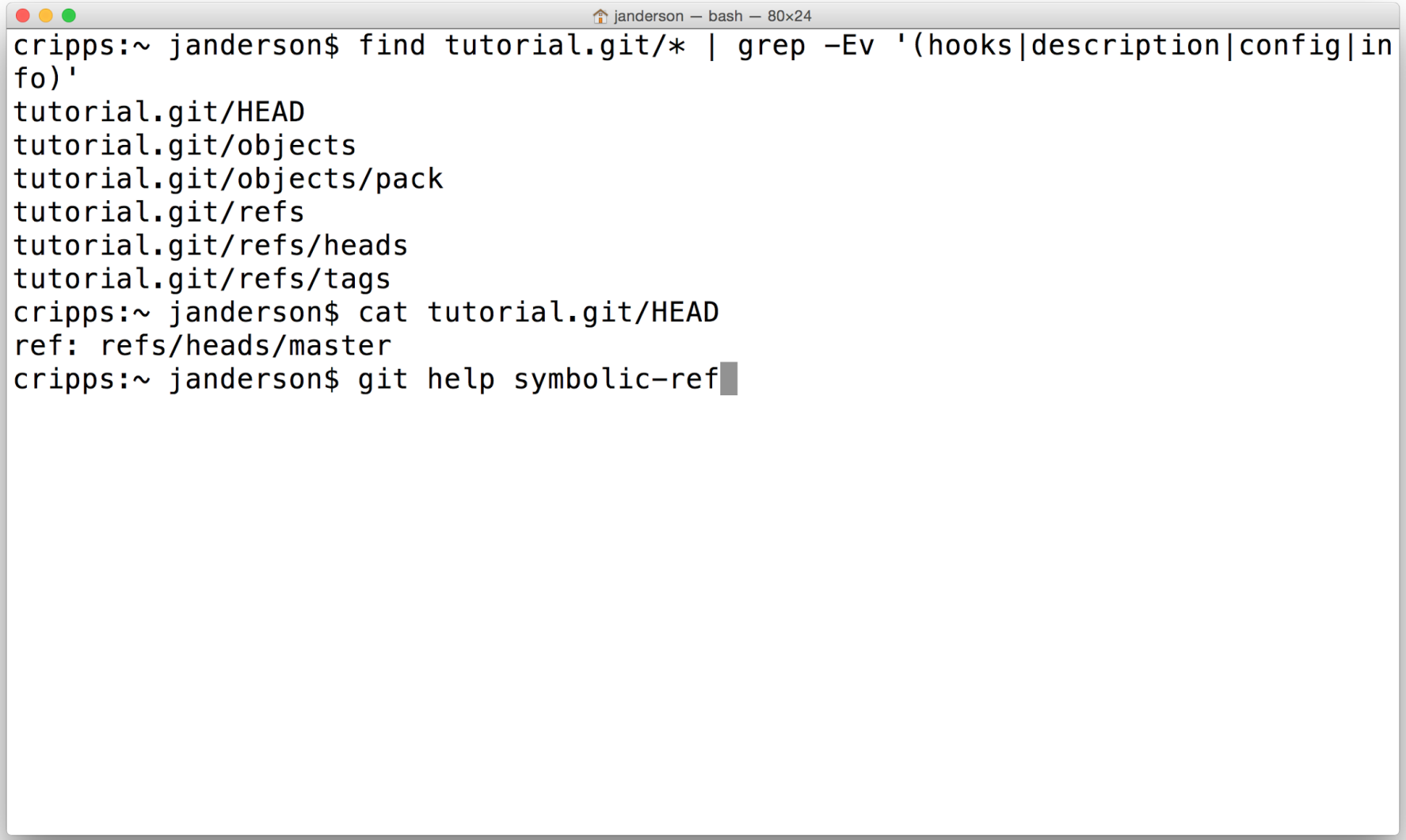

janderson — bash — 80×24
cripps:~ janderson$ find tutorial.git/* | grep -v 'hooks'
tutorial.git/HEAD
tutorial.git/config
tutorial.git/description
tutorial.git/info
tutorial.git/info/exclude
tutorial.git/objects
tutorial.git/objects/info
tutorial.git/objects/pack
tutorial.git/refs
tutorial.git/refs/heads
tutorial.git/refs/tags
cripps:~ janderson$ cat tutorial.git/description
Unnamed repository; edit this file 'description' to name the repository.
cripps:~ janderson$

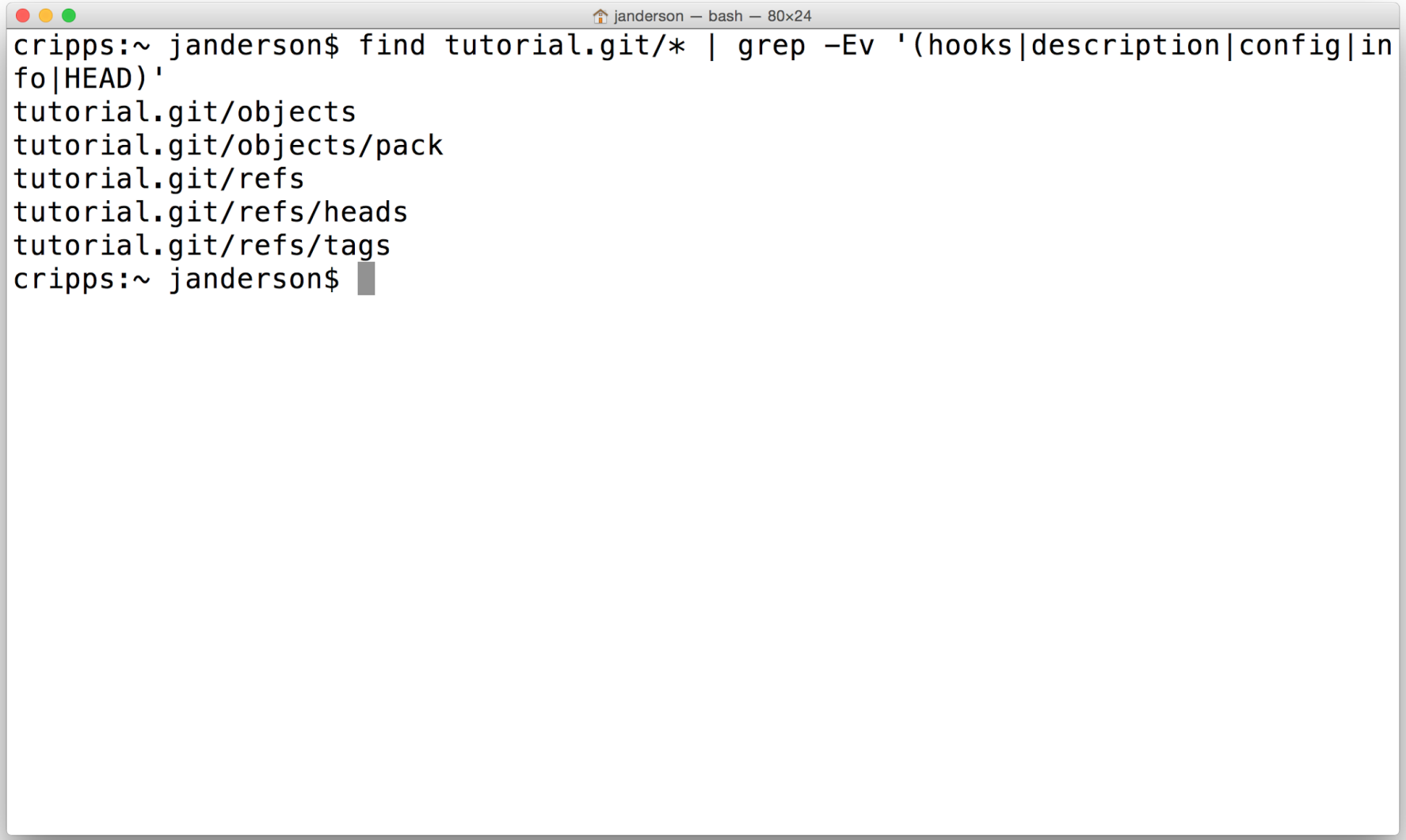

civilfritz.net/gitweb/
Go
+ Pocket
CU
CURC
civilfritz
Topics
Git tutorial
How To Make...ular Science
Topics pres...
Everyday Gi...
Google Slides
Distributed...
reddit: th
civilfritz.net Git
projects /
git
re
Search
List all projects
Project
Description
Owner
Last Change
bgbridge.git
Python ctypes interface to...
Jonathon Anderson
7 years ago
summary | shortlog | log | tree
bgenv.git
Query interface to the Blue...
Jonathon Anderson
3 years ago
summary | shortlog | log | tree
cbank.git
Resource allocation management...
Jonathon Anderson
5 years ago
summary | shortlog | log | tree
coddex.git
Relational databases for humans
Jonathon Anderson
4 months ago
summary | shortlog | log | tree
salt.git
Salt states for civilfritz
Jonathon Anderson
2 months ago
summary | shortlog | log | tree
wrapc.git
General-purpose binary wrapper...
Jonathon Anderson
5 years ago
summary | shortlog | log | tree
TXT
OPML

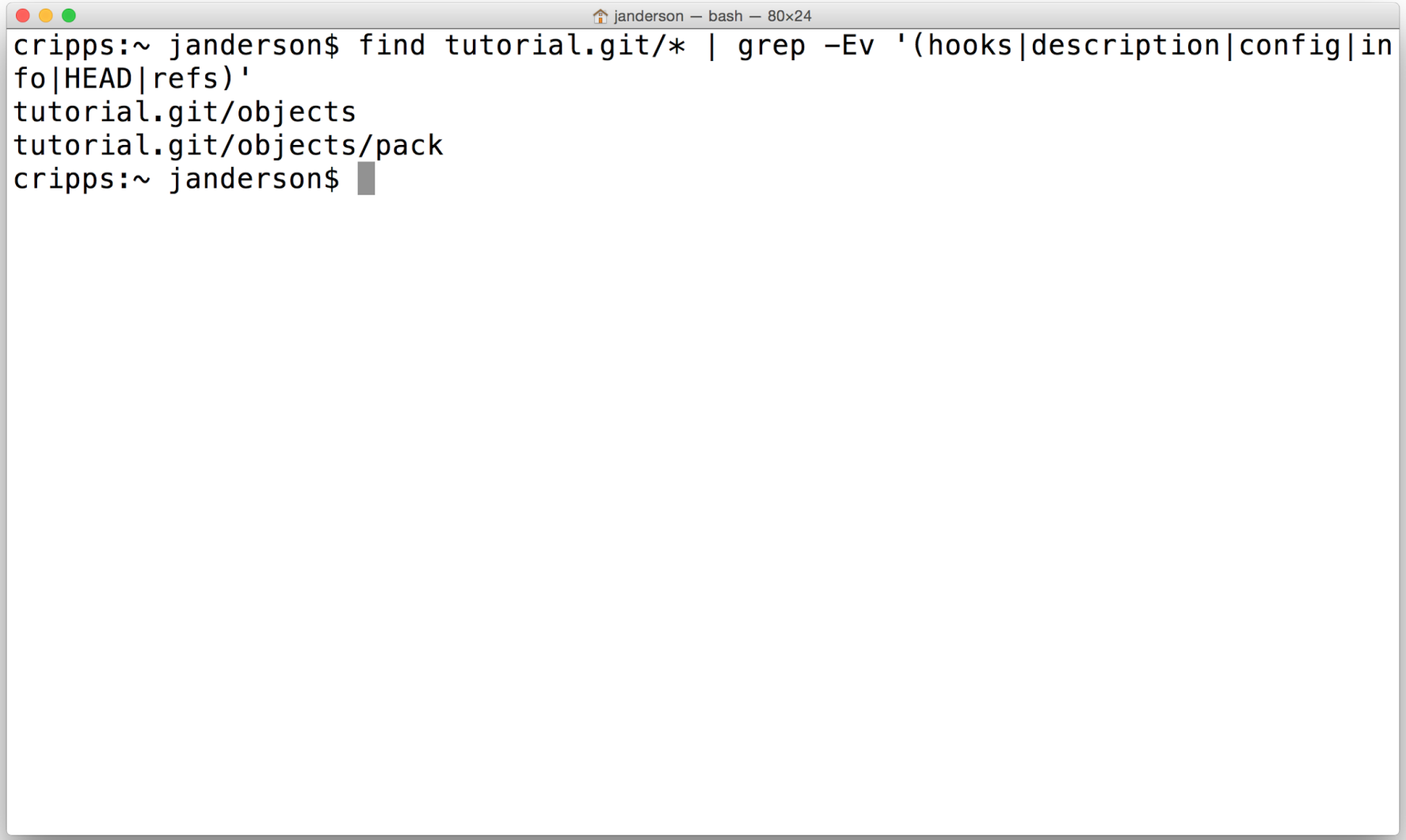

janderson — bash — 80×24
cripps:~ janderson$ find tutorial.git/* | grep -Ev '(hooks|description)'
tutorial.git/HEAD
tutorial.git/config
tutorial.git/info
tutorial.git/info/exclude
tutorial.git/objects
tutorial.git/objects/info
tutorial.git/objects/pack
tutorial.git/refs
tutorial.git/refs/heads
tutorial.git/refs/tags
cripps:~ janderson$ cat tutorial.git/config
[core]
repositoryformatversion = 0
filemode = true
bare = true
ignorecase = true
precomposeunicode = true
cripps:~ janderson$ git help config


janderson — bash — 80×24
cripps:~ janderson$ find tutorial.git/* | grep -Ev '(hooks|description|config)'
tutorial.git/HEAD
tutorial.git/info
tutorial.git/info/exclude
tutorial.git/objects
tutorial.git/objects/info
tutorial.git/objects/pack
tutorial.git/refs
tutorial.git/refs/heads
tutorial.git/refs/tags
cripps:~ janderson$ cat tutorial.git/info/exclude
git ls-files --others --exclude-from=.git/info/exclude
Lines that start with '#' are comments.
For a project mostly in C, the following would be a good set of
exclude patterns (uncomment them if you want to use them):
*.[oa]
*~
cripps:~ janderson$


janderson — bash — 80×24
cripps:~ janderson$ find tutorial.git/* | grep -Ev '(hooks|description|config)'
tutorial.git/HEAD
tutorial.git/info
tutorial.git/info/exclude
tutorial.git/objects
tutorial.git/objects/info
tutorial.git/objects/pack
tutorial.git/refs
tutorial.git/refs/heads
tutorial.git/refs/tags
cripps:~ janderson$ cat tutorial.git/info/exclude
git ls-files --others --exclude-from=.git/info/exclude
Lines that start with '#' are comments.
For a project mostly in C, the following would be a good set of
exclude patterns (uncomment them if you want to use them):
*.[oa]
*~
cripps:~ janderson$ echo "*.pyc" >>tutorial.git/info/exclude
cripps:~ janderson$


janderson — bash — 80×24
cripps:~ janderson$ find tutorial.git/* | grep -Ev '(hooks|description|config|info)'
tutorial.git/HEAD
tutorial.git/objects
tutorial.git/objects/pack
tutorial.git/refs
tutorial.git/refs/heads
tutorial.git/refs/tags
cripps:~ janderson$ cat tutorial.git/HEAD
ref: refs/heads/master
cripps:~ janderson$ git help symbolic-ref


janderson — bash — 80×24
cripps:~ janderson$ find tutorial.git/* | grep -Ev '(hooks|description|config|info|HEAD)'
tutorial.git/objects
tutorial.git/objects/pack
tutorial.git/refs
tutorial.git/refs/heads
tutorial.git/refs/tags
cripps:~ janderson$


janderson — bash — 80×24
cripps:~ janderson$ find tutorial.git/* | grep -Ev '(hooks|description|config|info|HEAD|refs)'
tutorial.git/objects
tutorial.git/objects/pack
cripps:~ janderson$

Terminology

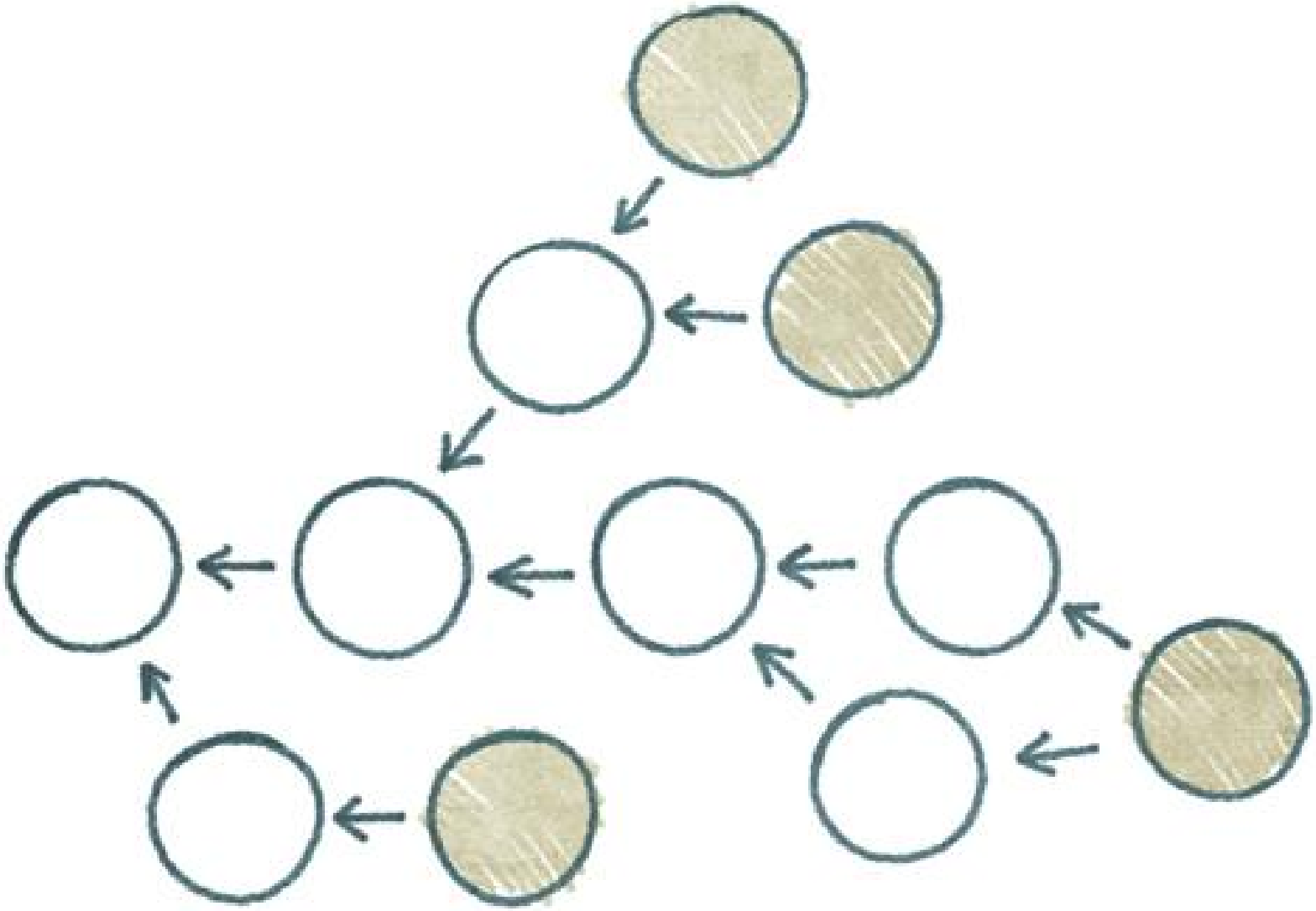
Repository:
A collection of commits, references, and configuration

Commit:
Point-in-time snapshot

Branch:
Directed, acyclic graph of commits

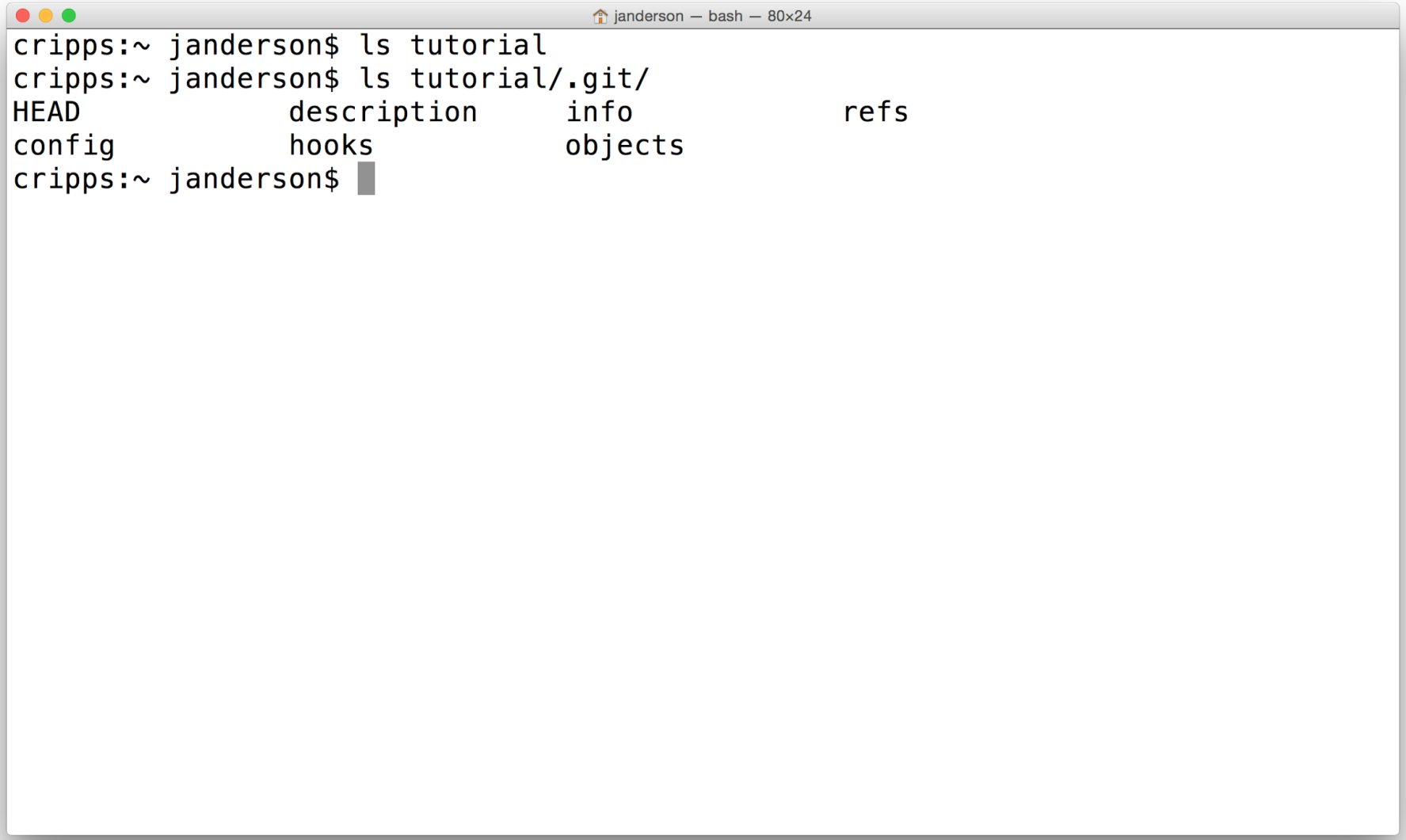
Head:
The last commit on a branch

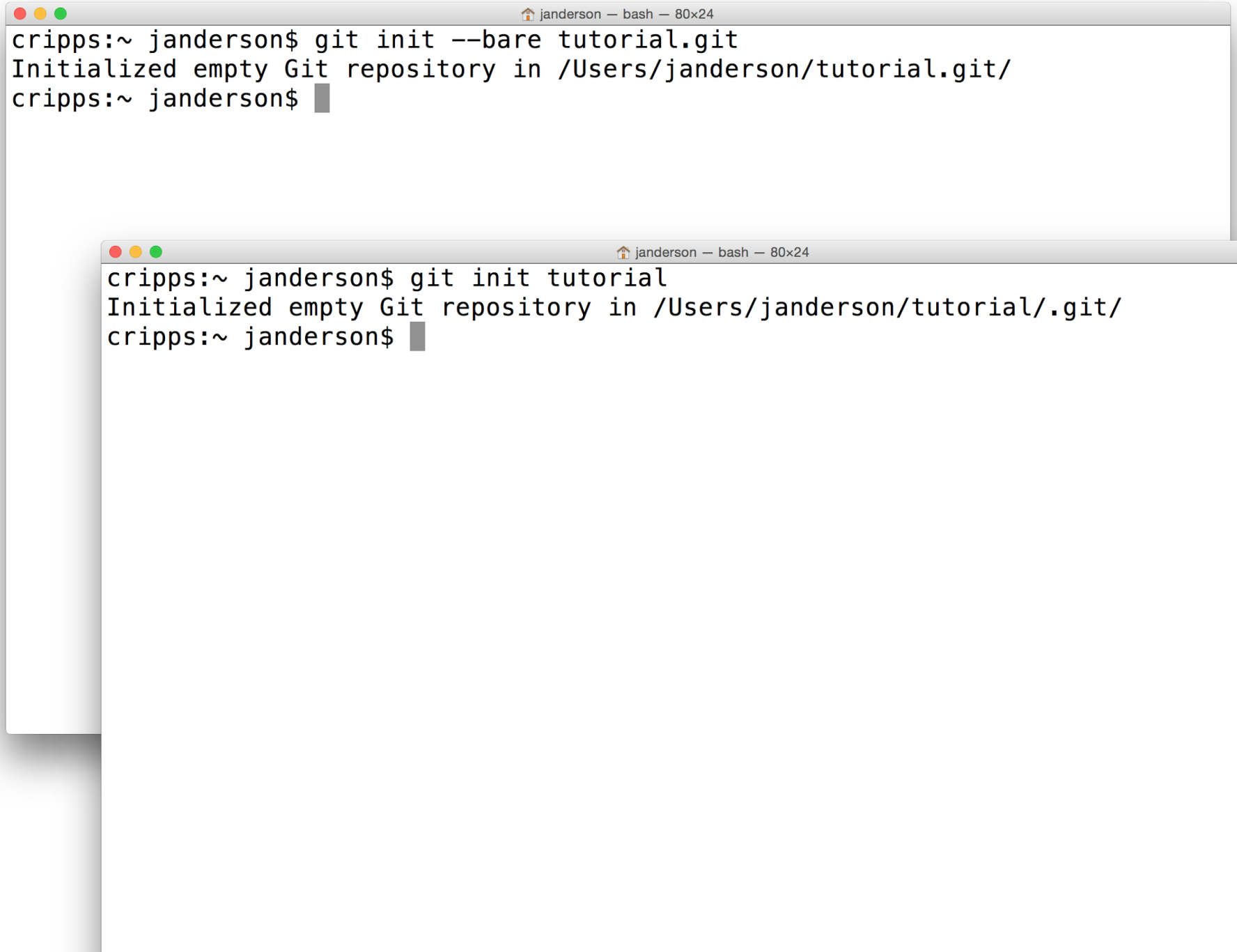
HEAD:
The last commit on the *current* branch

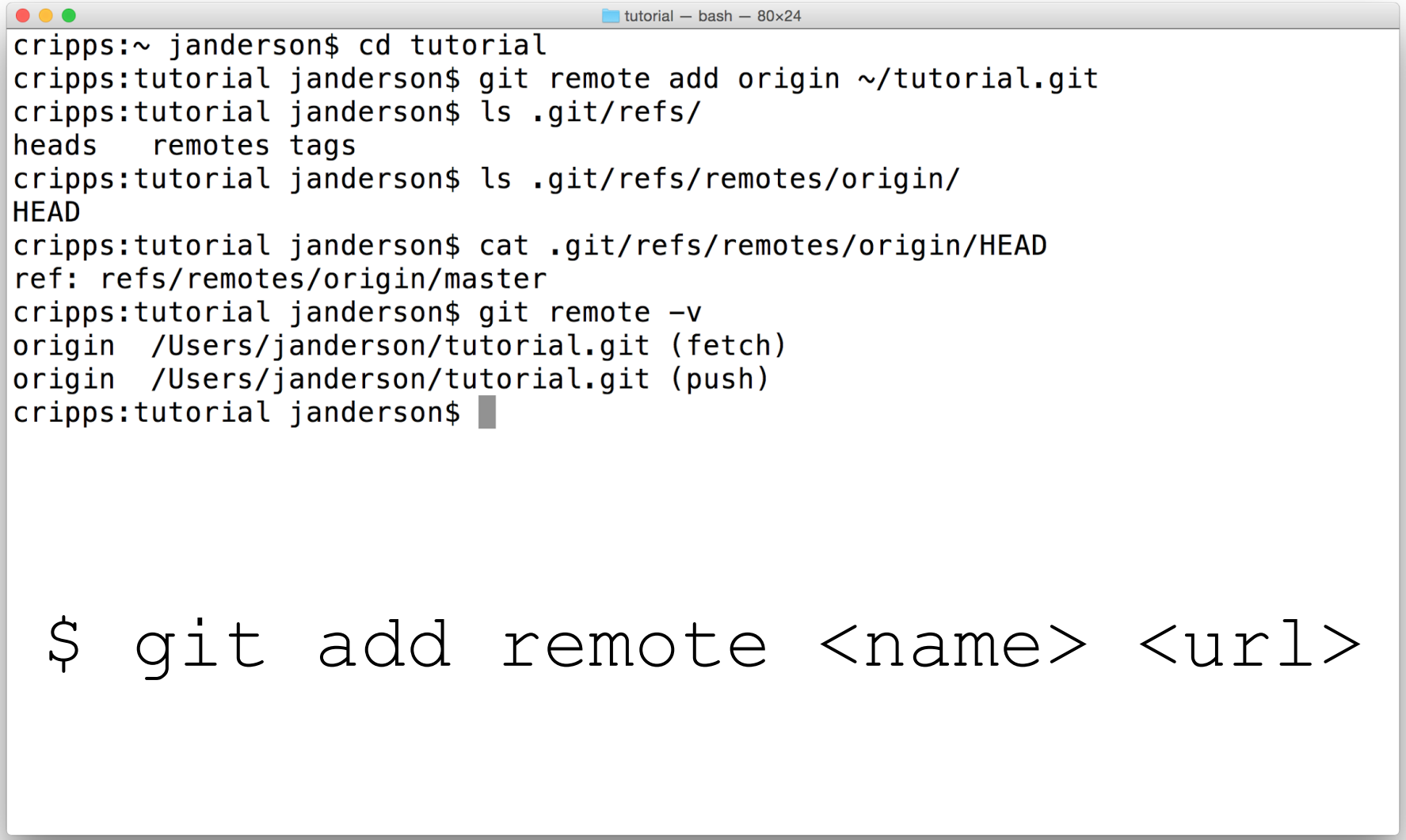


http://ericsink.com/vcbe/html/directed_acyclic_graphs.html

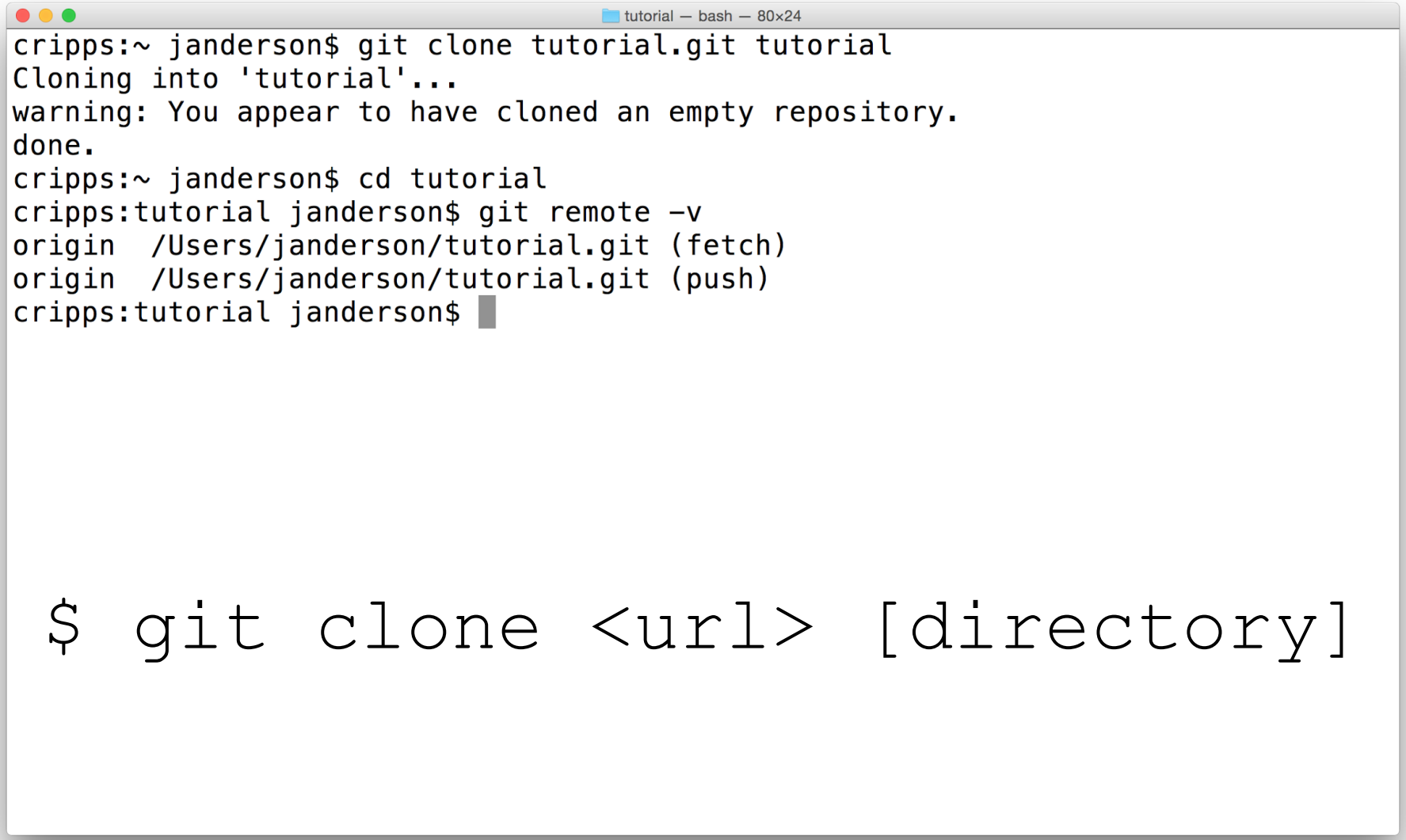

janderson — bash — 80×24
cripps:~ janderson$ git init tutorial
Initialized empty Git repository in /Users/janderson/tutorial/.git/
cripps:~ janderson$
$ git init <directory>


janderson — bash — 80×24
cripps:~ janderson$ ls tutorial
cripps:~ janderson$ ls tutorial/.git/
HEAD description info refs
config hooks objects
cripps:~ janderson$

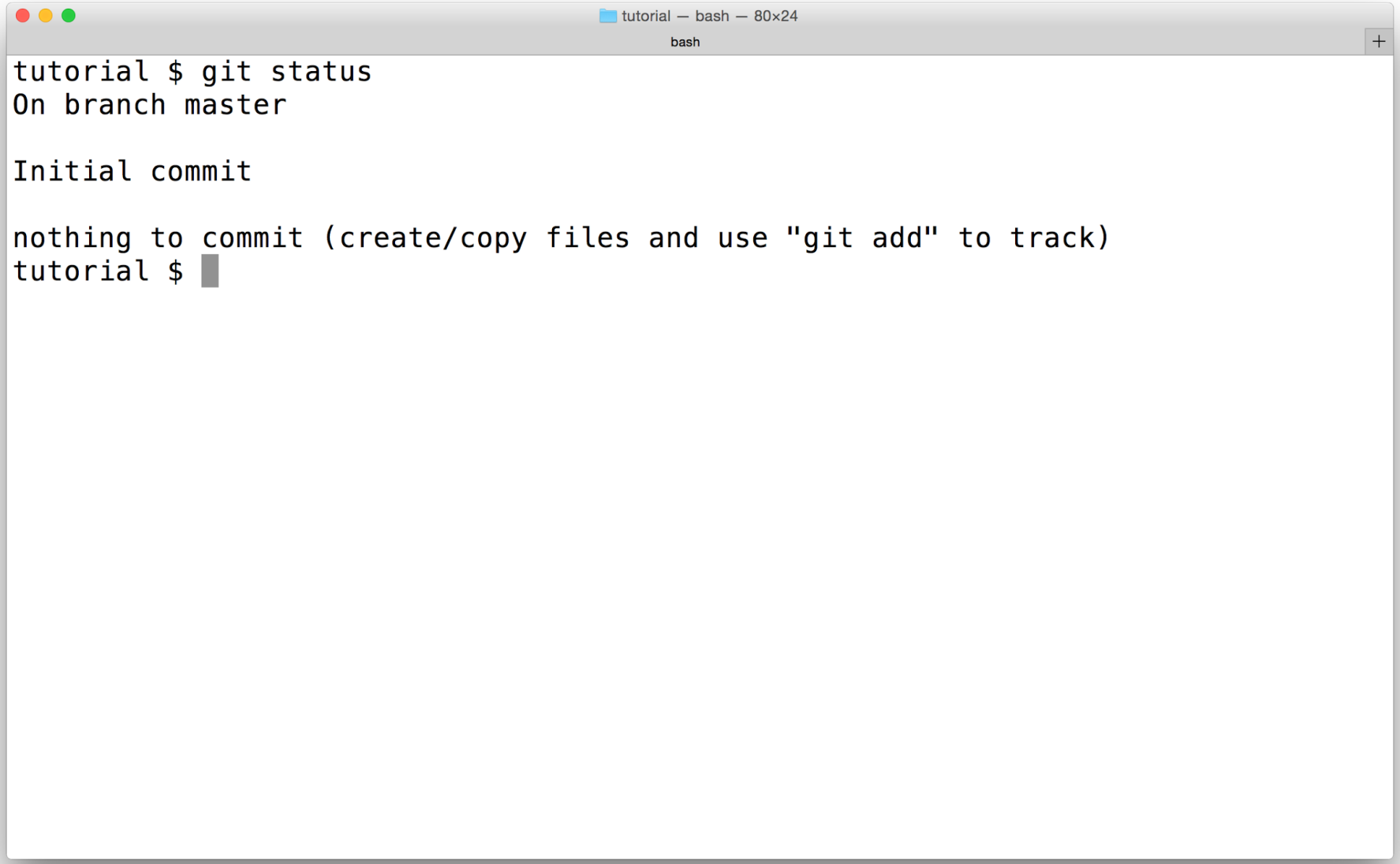

janderson — bash — 80×24
cripps:~ janderson$ git init --bare tutorial.git
Initialized empty Git repository in /Users/janderson/tutorial.git/
cripps:~ janderson$
janderson — bash — 80×24
cripps:~ janderson$ git init tutorial
Initialized empty Git repository in /Users/janderson/tutorial/.git/
cripps:~ janderson$

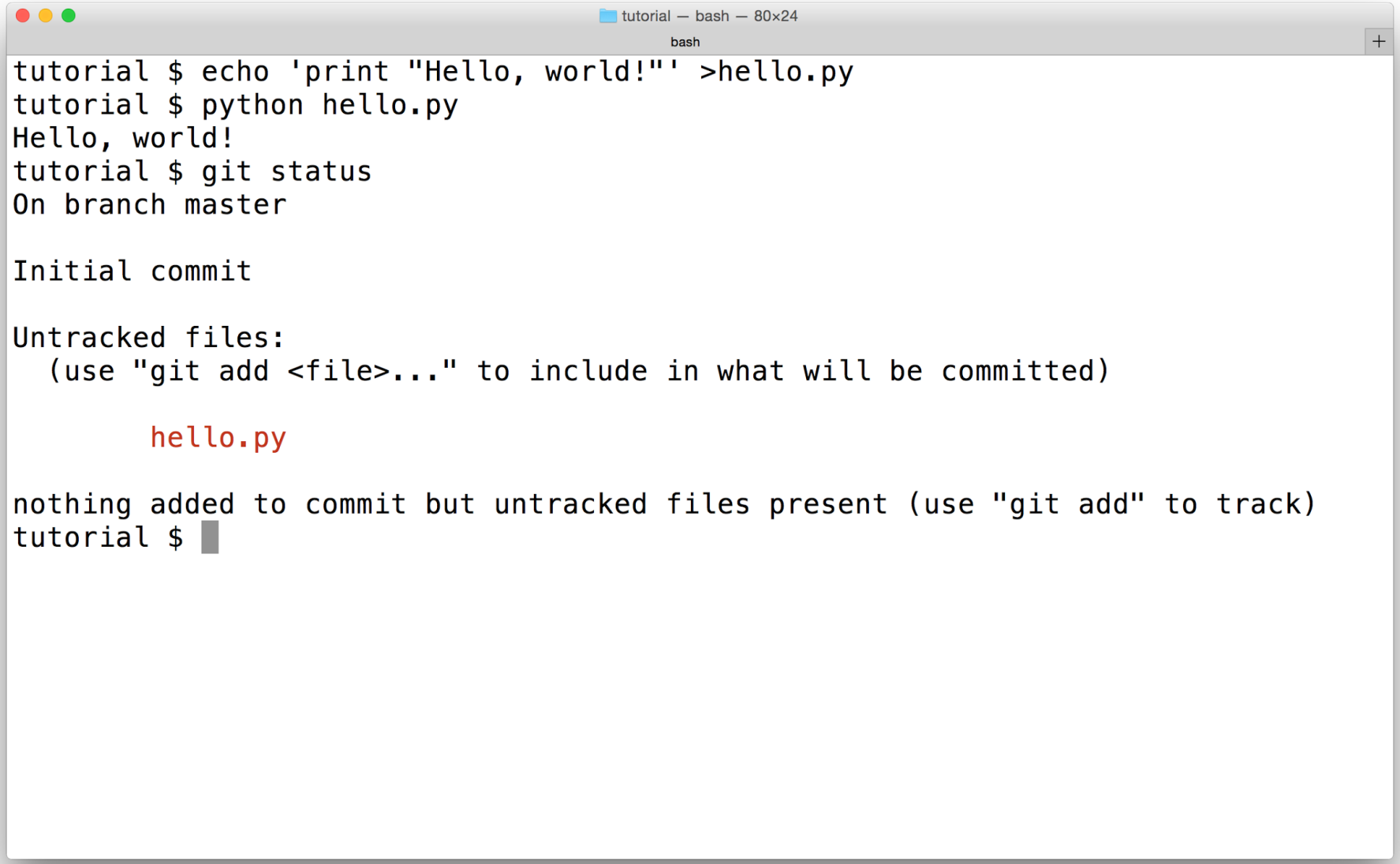


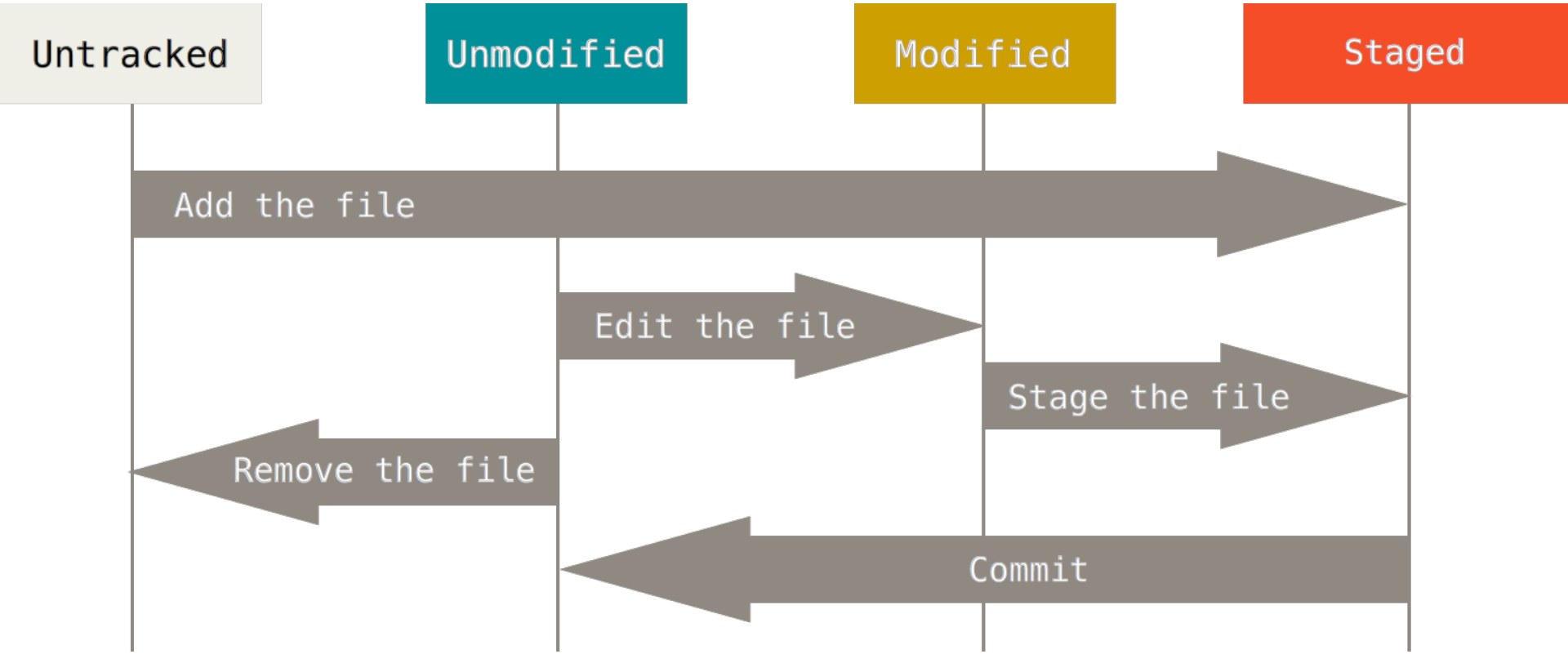
$ git add remote <name> <url>


tutorial — bash — 80×24
cripps:~ janderson$ git clone tutorial.git tutorial
Cloning into 'tutorial'...
warning: You appear to have cloned an empty repository.
done.
cripps:~ janderson$ cd tutorial
cripps:tutorial janderson$ git remote -v
origin /Users/janderson/tutorial.git (fetch)
origin /Users/janderson/tutorial.git (push)
cripps:tutorial janderson$
$ git clone <url> [directory]

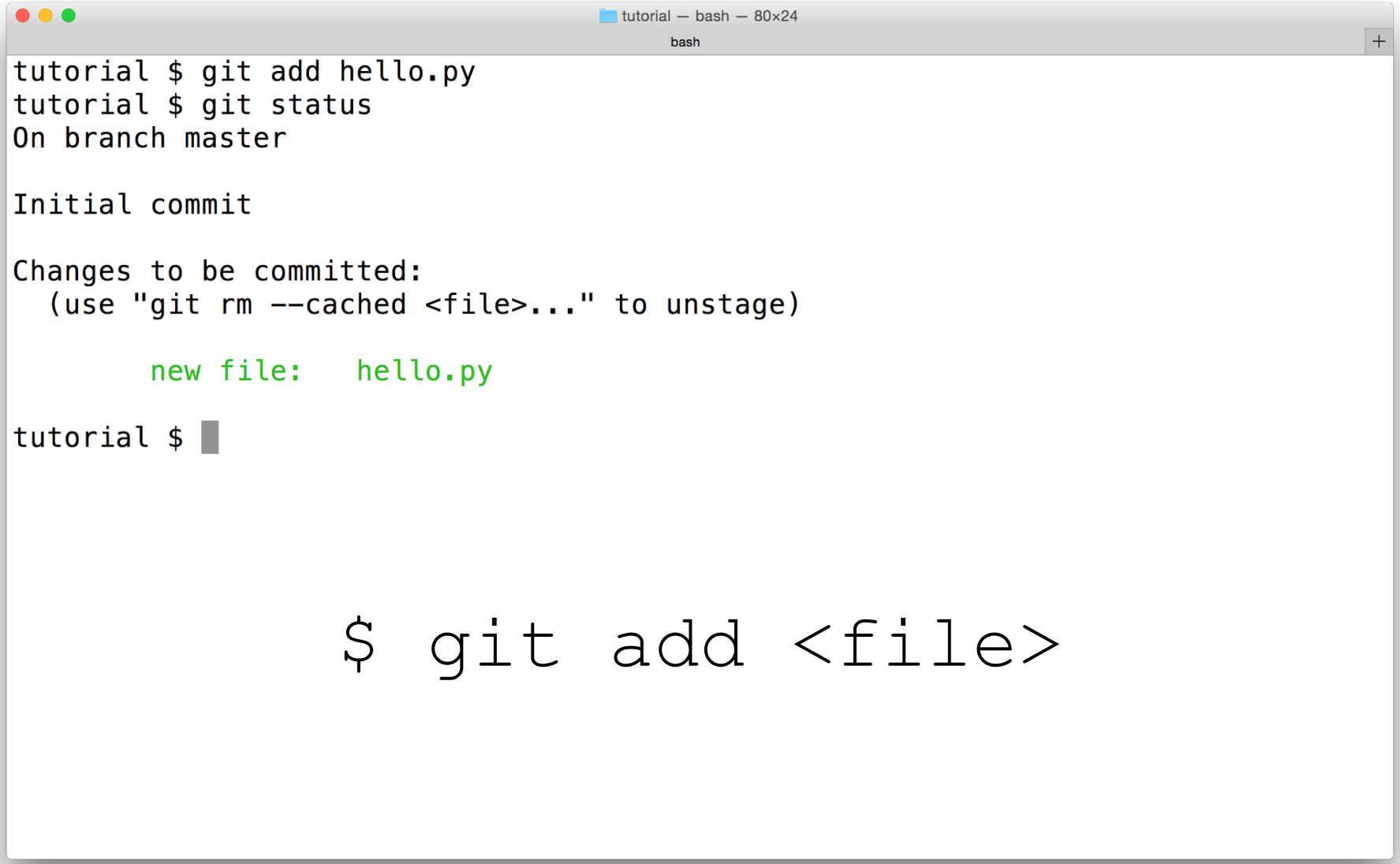
Adding and committing content


tutorial — bash — 80×24
bash
tutorial $ git status
On branch master
Initial commit
nothing to commit (create/copy files and use "git add" to track)
tutorial $

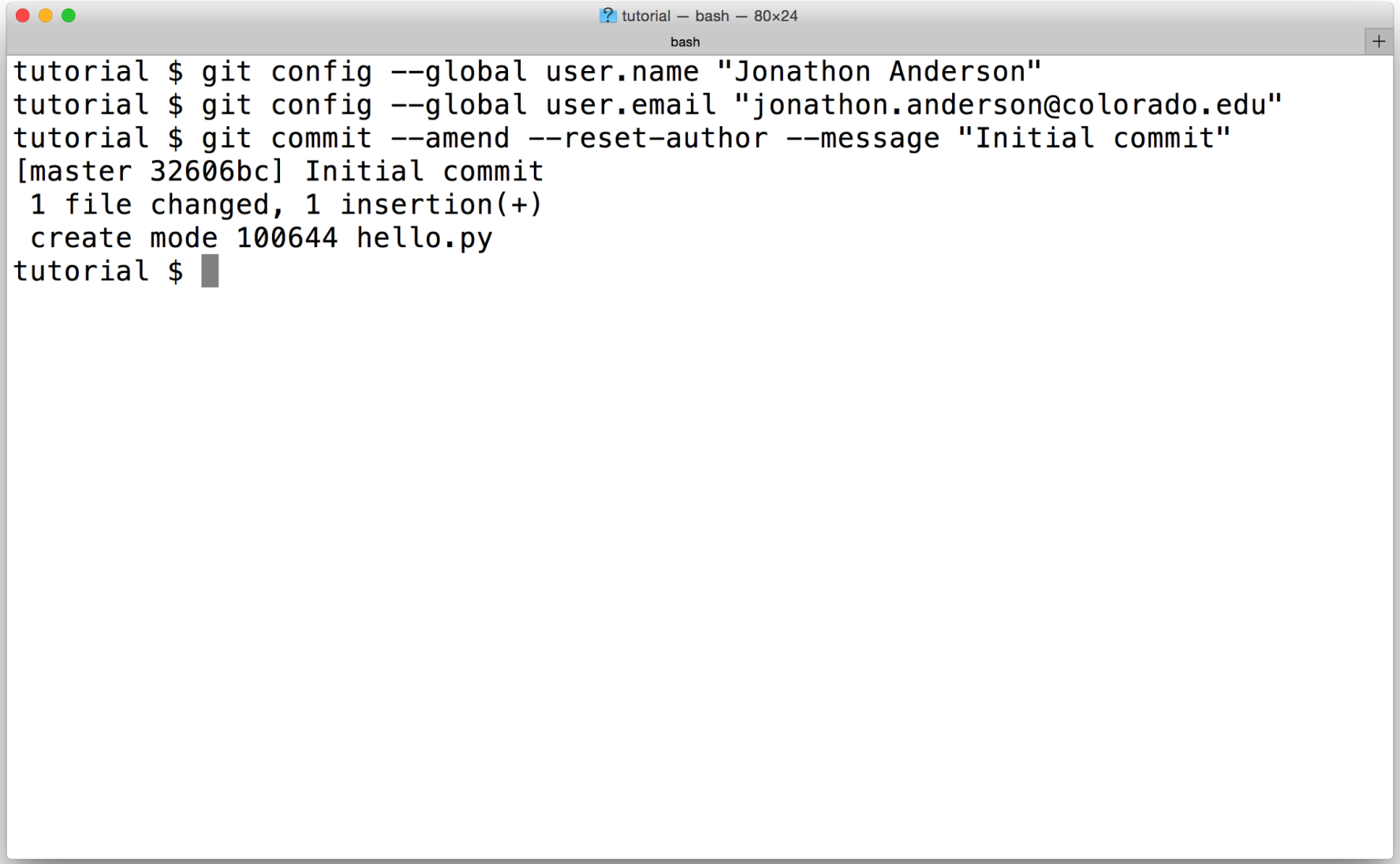

tutorial — bash — 80×24
bash
tutorial $ echo 'print "Hello, world!"' >hello.py
tutorial $ python hello.py
Hello, world!
tutorial $ git status
On branch master
Initial commit
Untracked files:
(use "git add <file>..." to include in what will be committed)
hello.py
nothing added to commit but untracked files present (use "git add" to track)
tutorial $

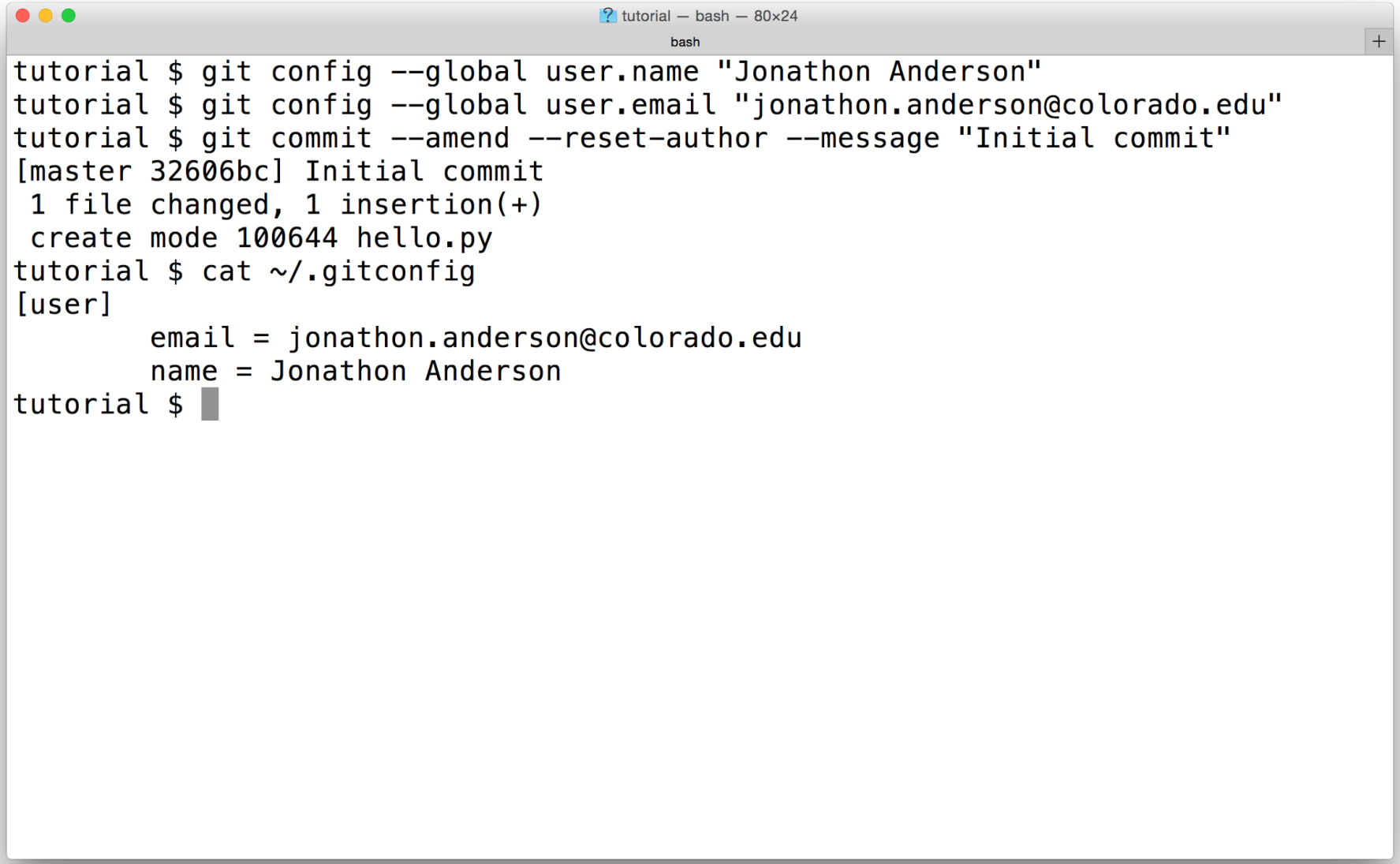


http://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository


tutorial — bash — 80×24
bash
tutorial $ git add hello.py
tutorial $ git status
On branch master
Initial commit
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
new file: hello.py
tutorial $
$ git add <file>


tutorial — bash — 80×24
bash
tutorial $ git commit -m "Initial commit"
[master (root-commit) ab20061] Initial commit
 Committer: Jonathon Anderson (CU) <joan5896@rgnt1-6-95-dhcp.int.colorado.edu>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:
git config --global user.name "Your Name"
git config --global user.email you@example.com
After doing this, you may fix the identity used for this commit with:
git commit --amend --reset-author
1 file changed, 1 insertion(+)
create mode 100644 hello.py
tutorial $
$ git commit -m <message>


tutorial — bash — 80×24
bash
tutorial $ git config --global user.name "Jonathon Anderson"
tutorial $ git config --global user.email "jonathon.anderson@colorado.edu"
tutorial $ git commit --amend --reset-author --message "Initial commit"
[master 32606bc] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 hello.py
tutorial $


tutorial — bash — 80×24
bash
tutorial $ git config --global user.name "Jonathon Anderson"
tutorial $ git config --global user.email "jonathon.anderson@colorado.edu"
tutorial $ git commit --amend --reset-author --message "Initial commit"
[master 32606bc] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 hello.py
tutorial $ cat ~/.gitconfig
[user]
        email = jonathon.anderson@colorado.edu
        name = Jonathon Anderson
tutorial $

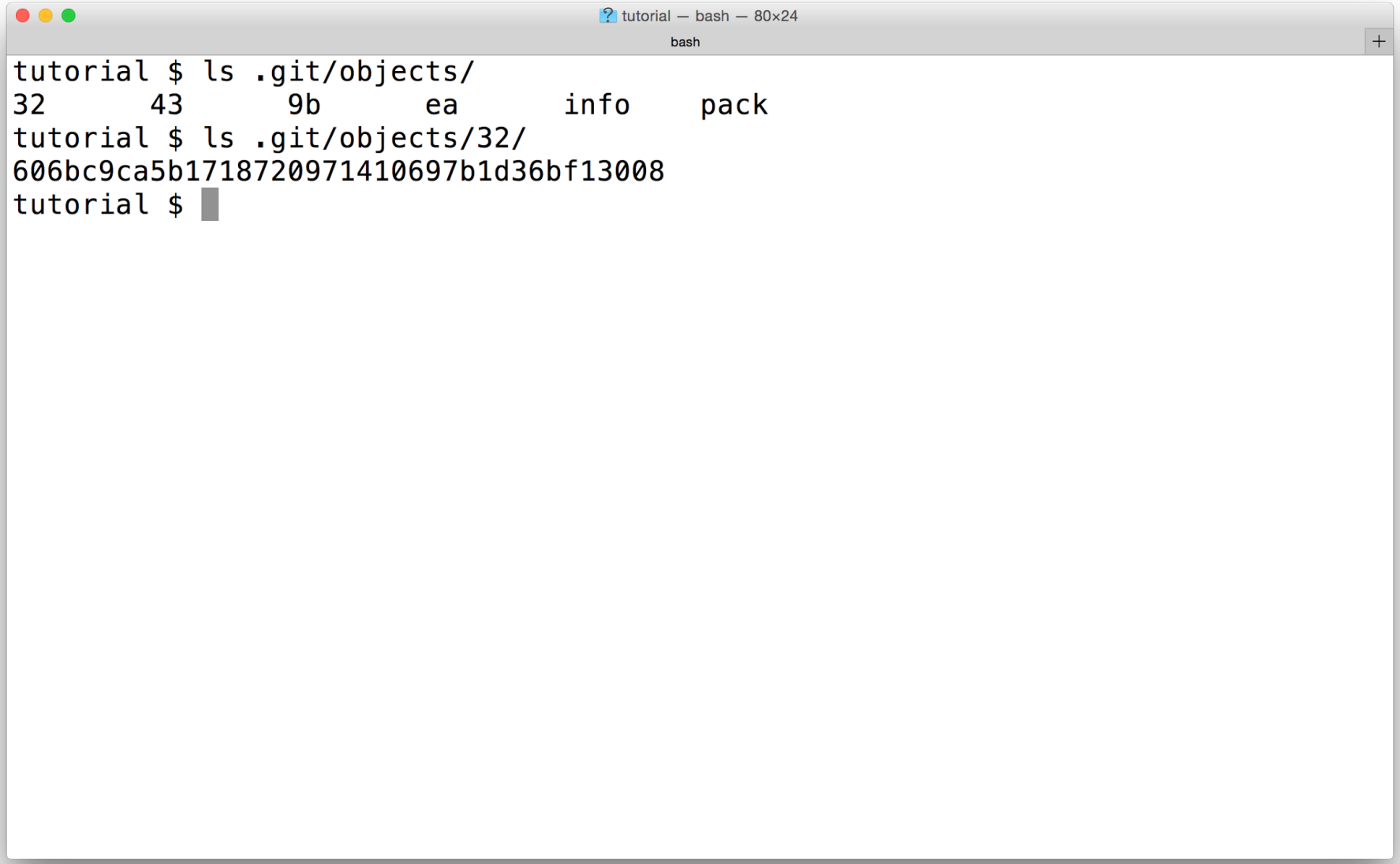
The SHA(-1 hash)

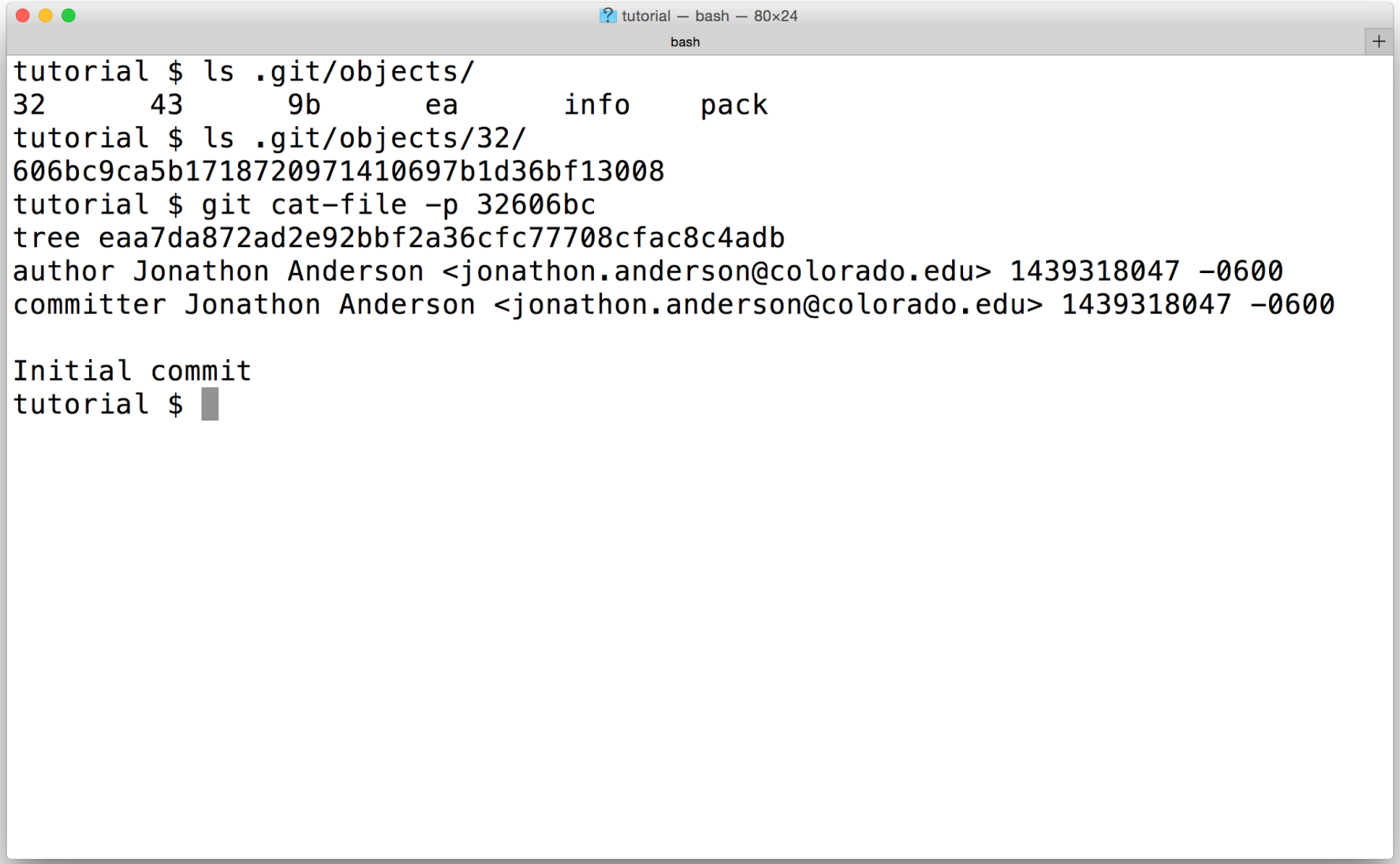
The SHA

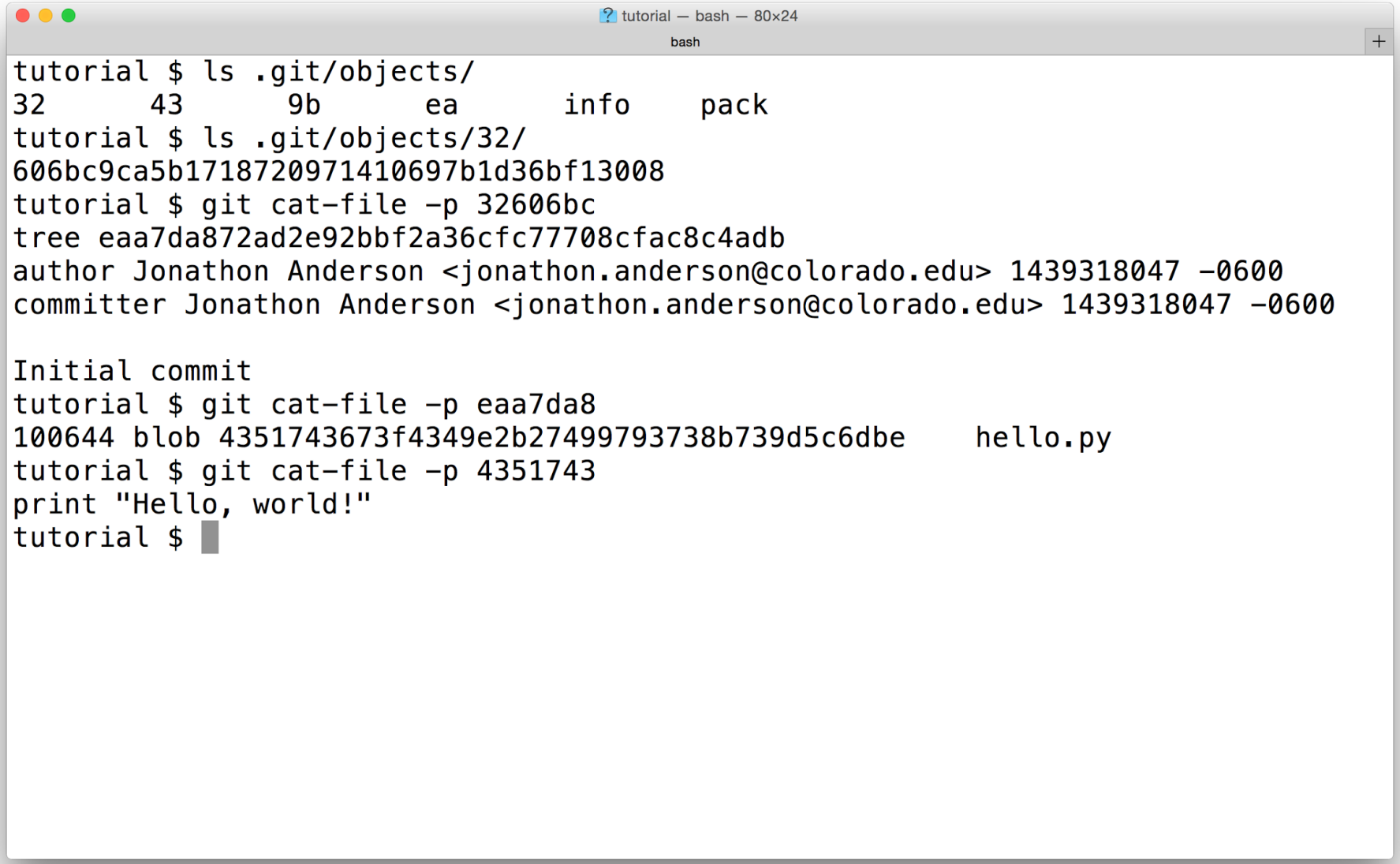
32606bc9ca5b1718720971410697b1d36bf13008

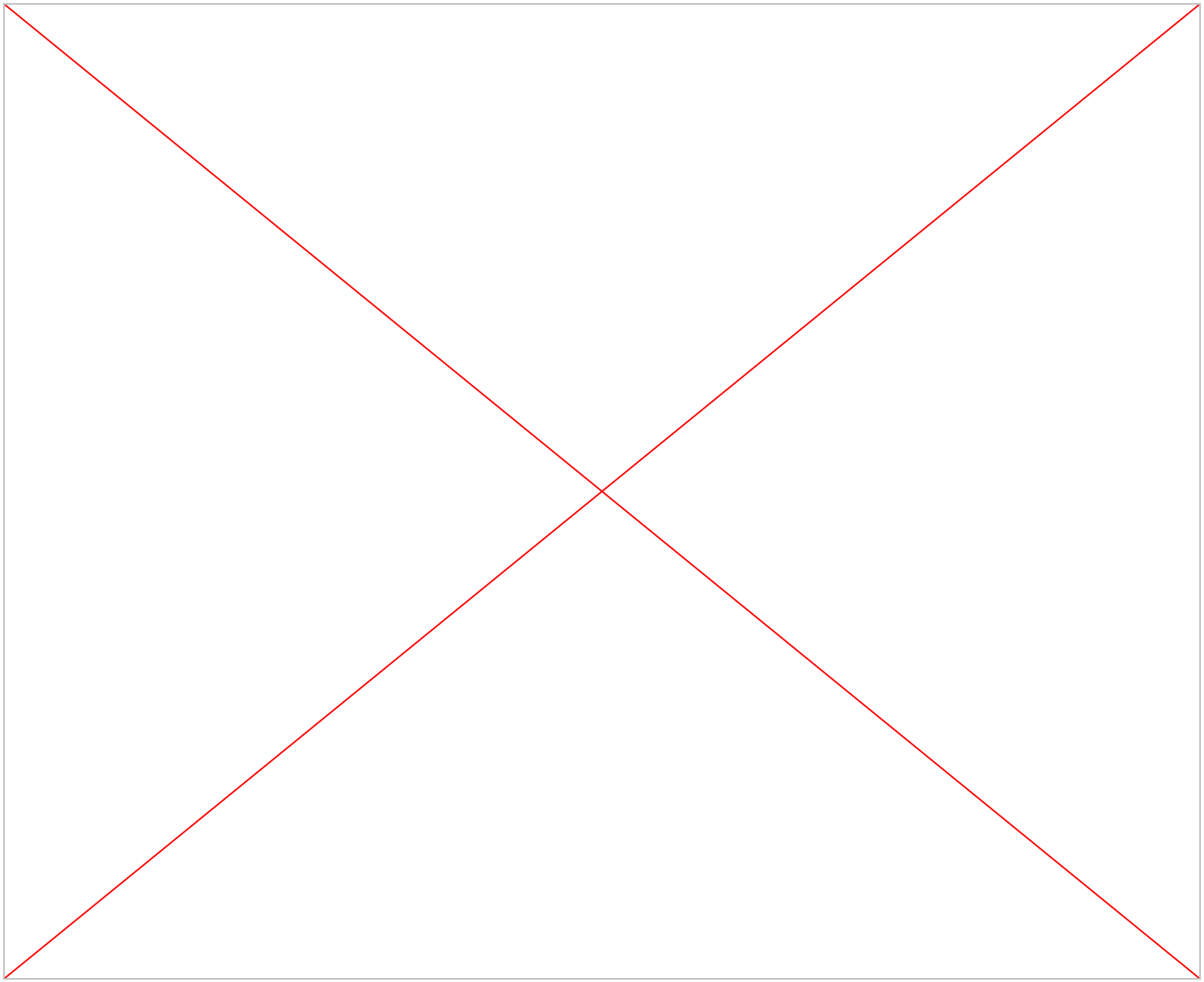
The SHA

32606bc9ca5b1718720971410697b1d36bf13008


tutorial — bash — 80×24
bash
tutorial $ ls .git/objects/
32 43 9b ea info pack
tutorial $ ls .git/objects/32/
606bc9ca5b17187209714106 97b1d36bf13008
tutorial $


tutorial — bash — 80×24
bash
tutorial $ ls .git/objects/
32 43 9b ea info pack
tutorial $ ls .git/objects/32/
606bc9ca5b1718720971410697b1d36bf13008
tutorial $ git cat-file -p 32606bc
tree eaa7da872ad2e92bbf2a36cfc77708cfac8c4adb
author Jonathon Anderson <jonathon.anderson@colorado.edu> 1439318047 -0600
committer Jonathon Anderson <jonathon.anderson@colorado.edu> 1439318047 -0600
Initial commit
tutorial $


tutorial — bash — 80×24
bash
tutorial $ ls .git/objects/
32 43 9b ea info pack
tutorial $ ls .git/objects/32/
606bc9ca5b1718720971410697b1d36bf13008
tutorial $ git cat-file -p 32606bc
tree eaa7da872ad2e92bbf2a36cfc77708cfac8c4adb
author Jonathon Anderson <jonathon.anderson@colorado.edu> 1439318047 -0600
committer Jonathon Anderson <jonathon.anderson@colorado.edu> 1439318047 -0600
Initial commit
tutorial $ git cat-file -p eaa7da8
100644 blob 4351743673f4349e2b27499793738b739d5c6dbe hello.py
tutorial $ git cat-file -p 4351743
print "Hello, world!"
tutorial $



https://schacon.github.io/gitbook/1_the_git_object_model.html

The SHA: why it’s awesome

- Git can quickly determine whether two objects are identical or not, just by comparing names.

The SHA: why it’s awesome

- Git can quickly determine whether two objects are identical or not, just by comparing names.
- Since object names are computed the same way in every repository, the same content stored in two repositories will always be stored under the same name.

The SHA: why it’s awesome

- Git can quickly determine whether two objects are identical or not, just by comparing names.
- Since object names are computed the same way in every repository, the same content stored in two repositories will always be stored under the same name.
- For the same reason, different content created in two different repositories will never have conflicting names.

The SHA: why it's awesome

- Git can quickly determine whether two objects are identical or not, just by comparing names.
- Since object names are computed the same way in every repository, the same content stored in two repositories will always be stored under the same name.
- For the same reason, different content created in two different repositories will never have conflicting names.

Because math.

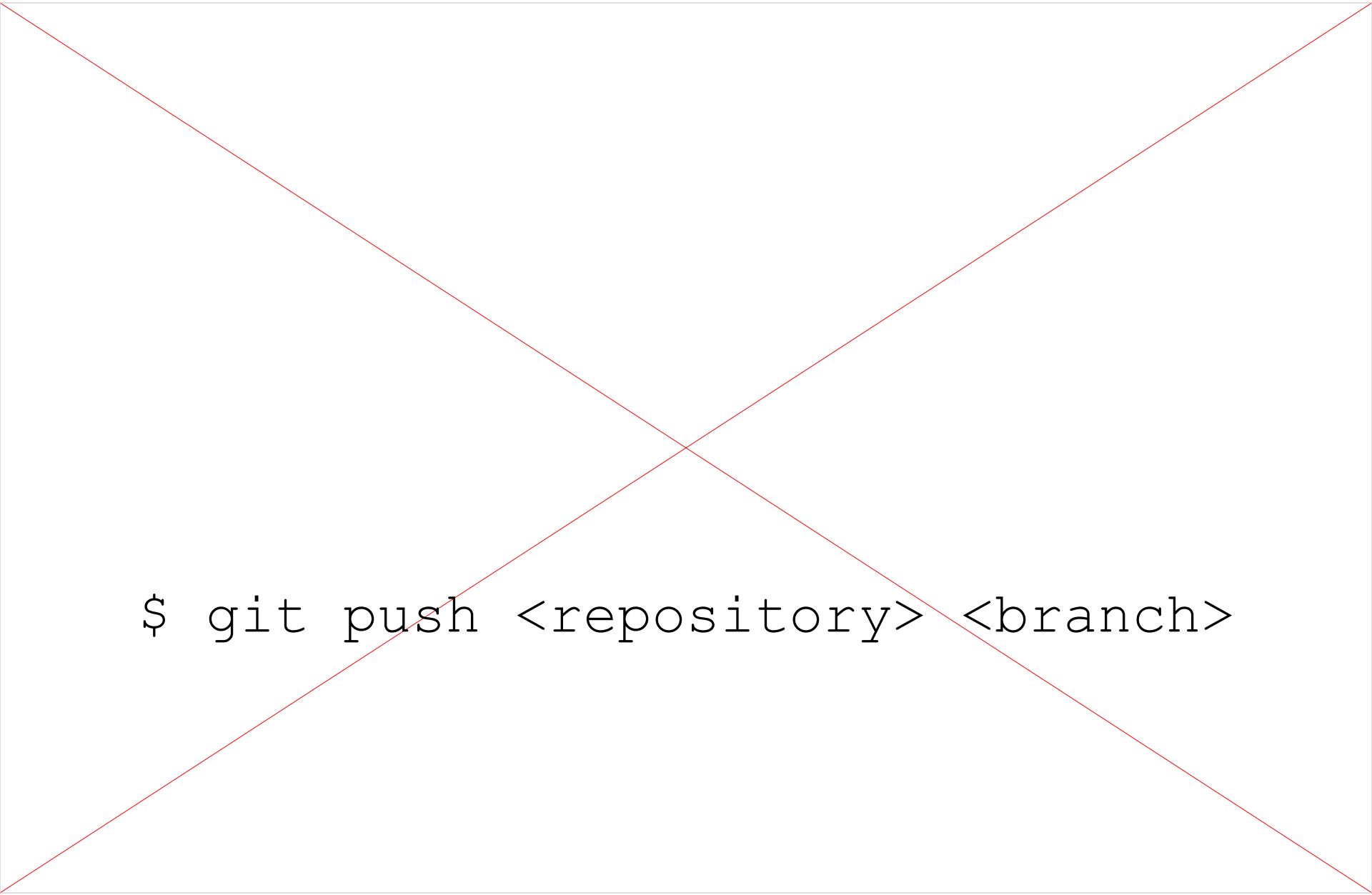
The SHA: why it’s awesome

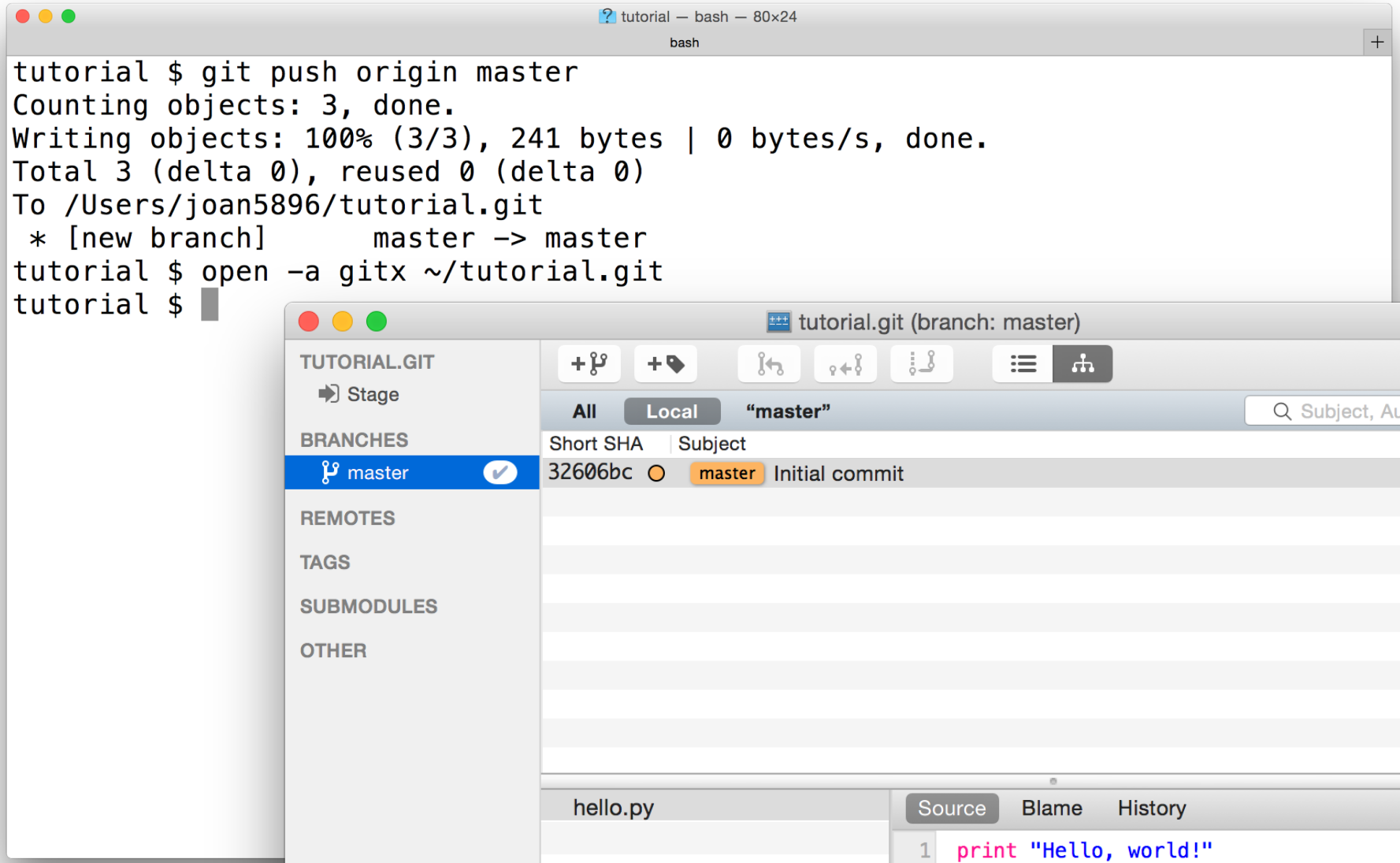
- Git can quickly determine whether two objects are identical or not, just by comparing names.
- Since object names are computed the same way in every repository, the same content stored in two repositories will always be stored under the same name.
- For the same reason, different content created in two different repositories will never have conflicting names.
- Git can detect errors when it reads an object by checking that the object's name is still the SHA1 hash of its contents.

The SHA: why it’s awesome

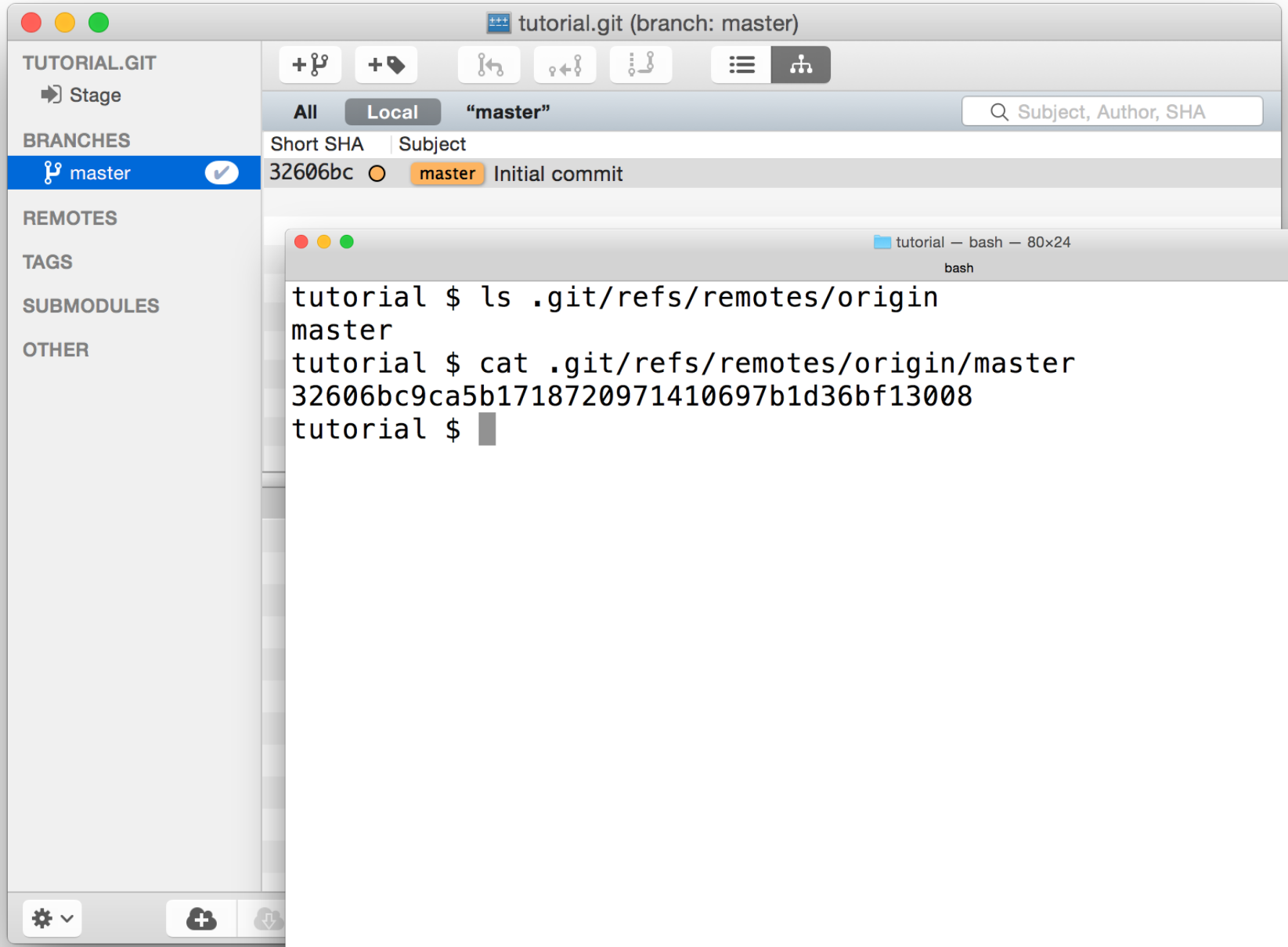
- Git can quickly determine whether two objects are identical or not, just by comparing names.
- Since object names are computed the same way in every repository, the same content stored in two repositories will always be stored under the same name.
- For the same reason, different content created in two different repositories will never have conflicting names.
- Git can detect errors when it reads an object by checking that the object's name is still the SHA1 hash of its contents.

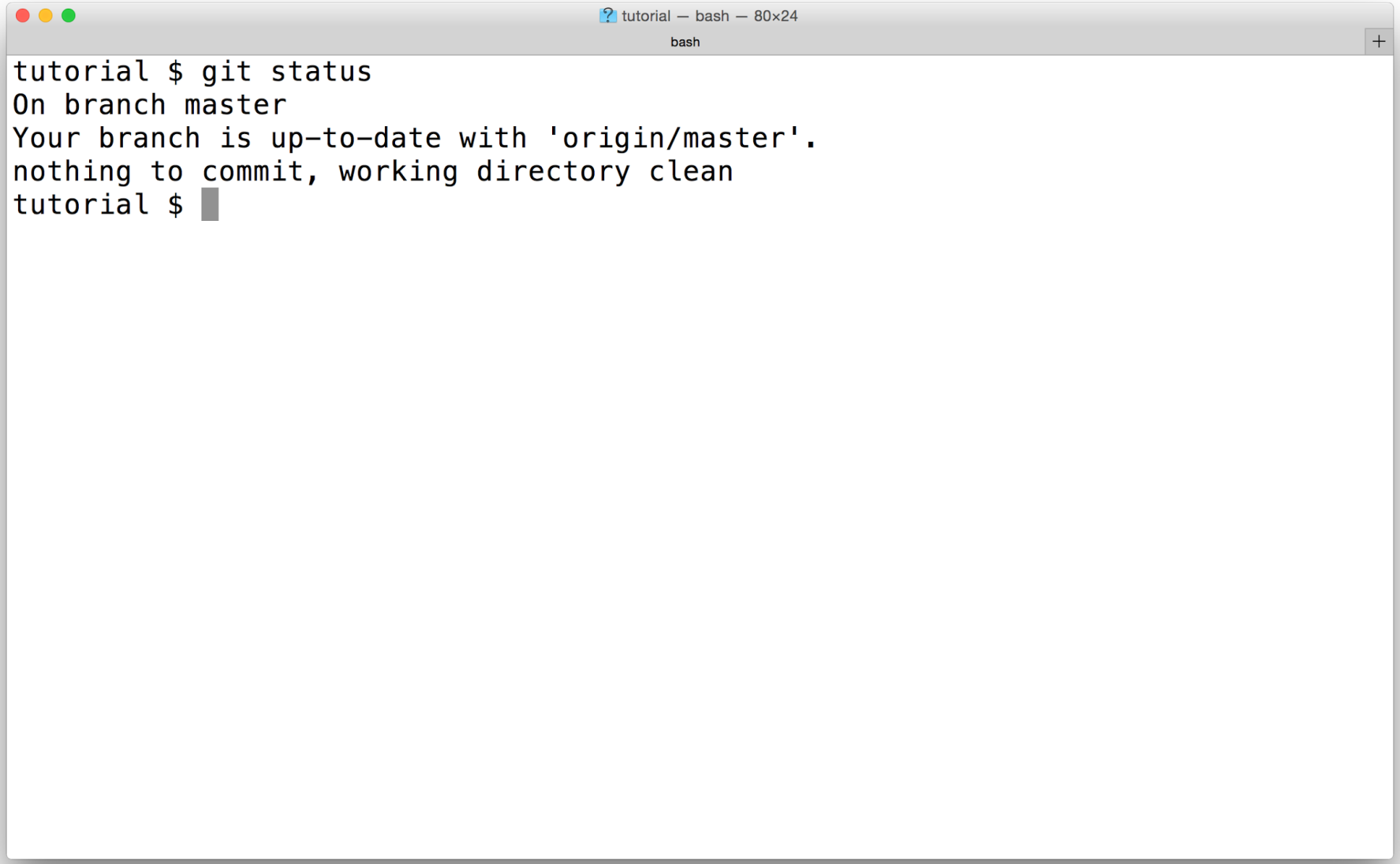
```
$ git fsck
```

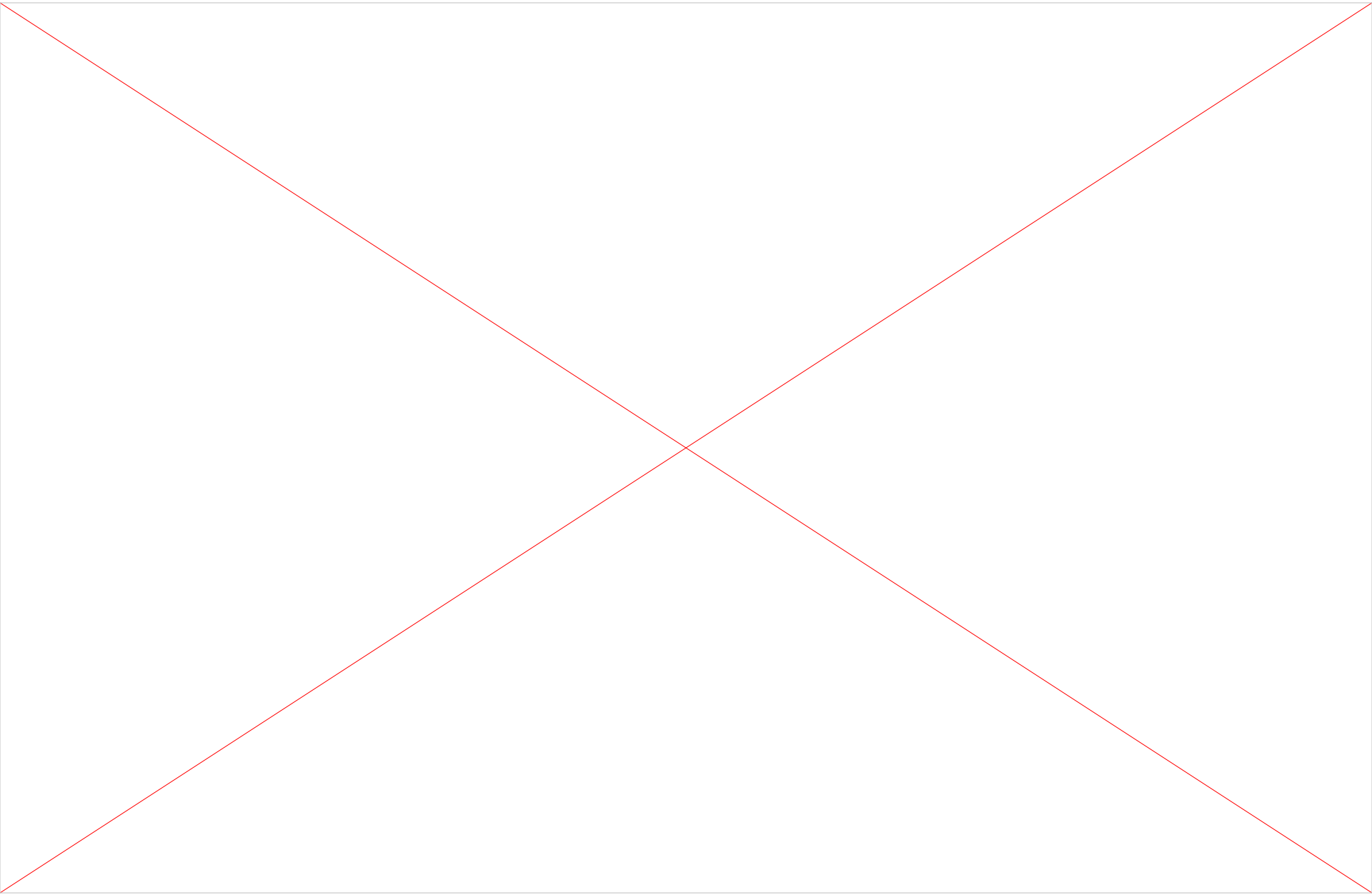

$ git push <repository> <branch>

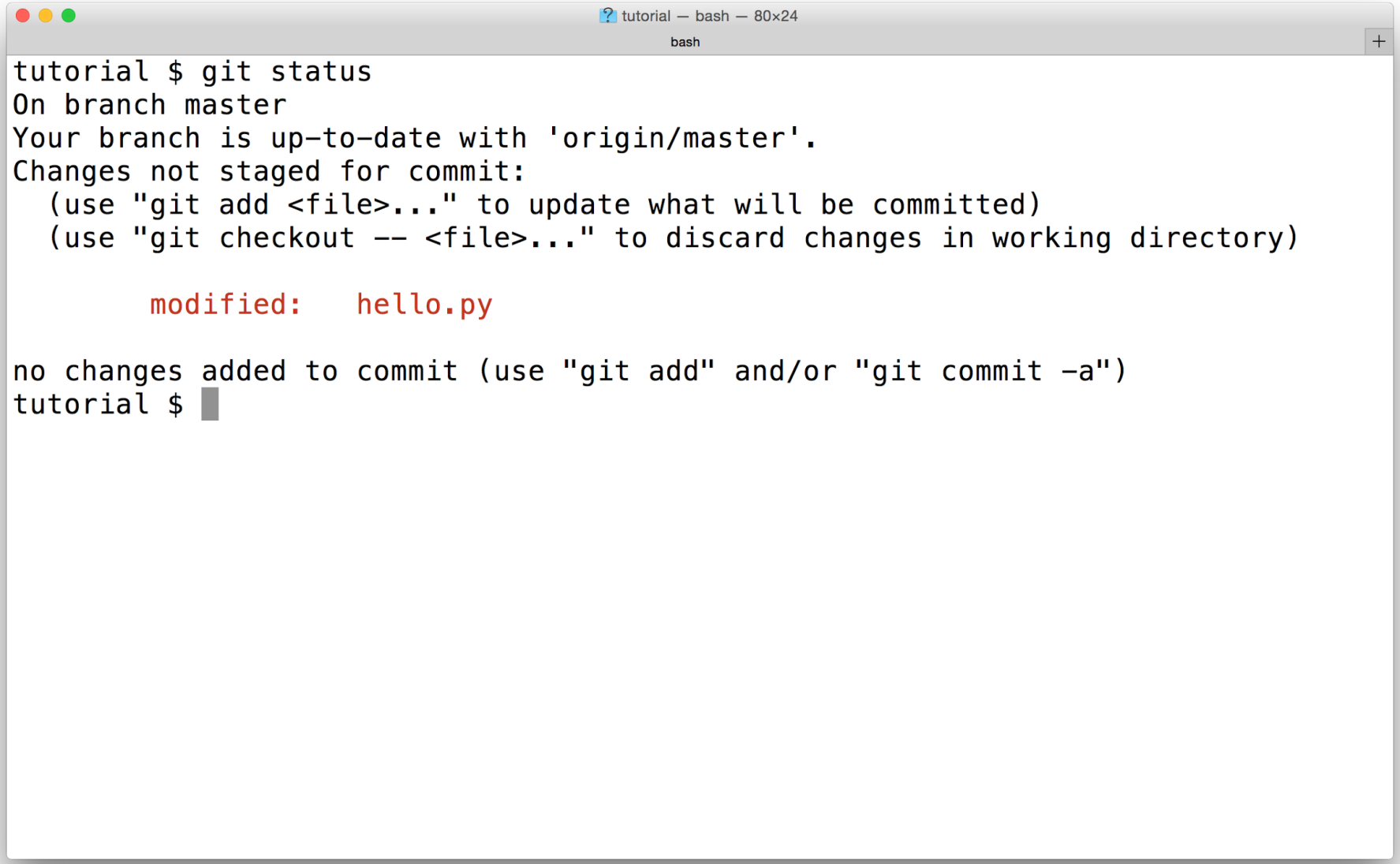


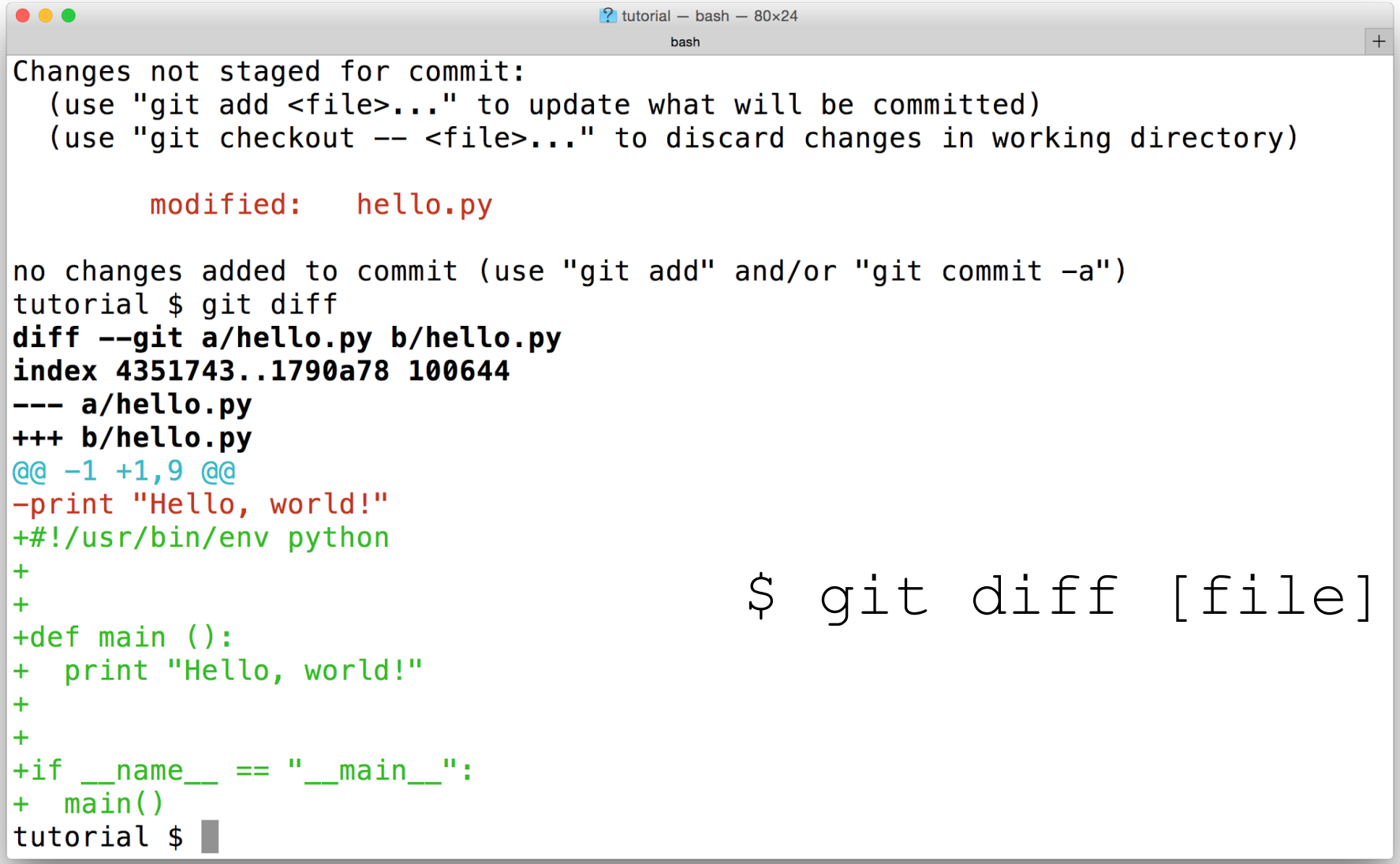
http://rowanj.github.io/gitx/

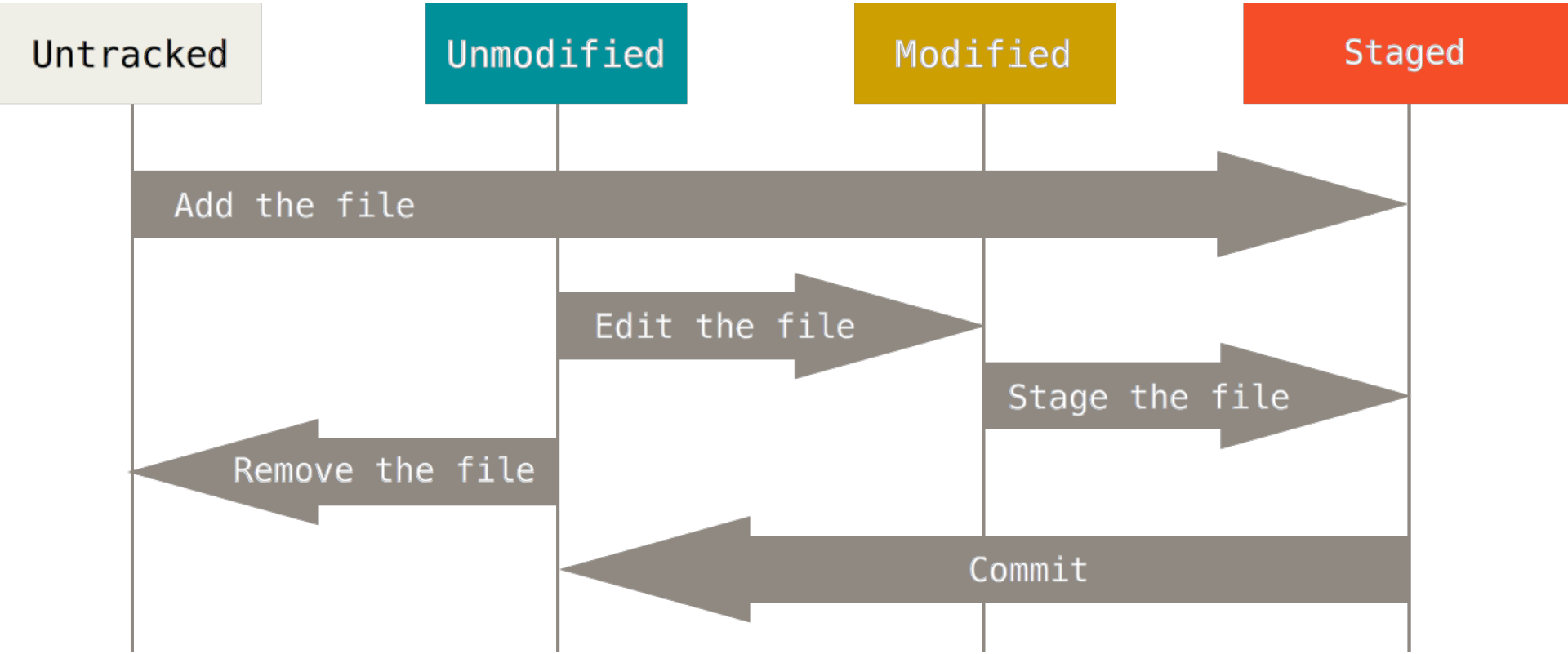

tutorial.git (branch: master)
TUTORIAL.GIT
Stage
BRANCHES
master
REMOTES
TAGS
SUBMODULES
OTHER
All
Local
“master”
Subject, Author, SHA
Short SHA
Subject
32606bc
master
Initial commit
tutorial — bash — 80×24
bash
tutorial $ ls .git/refs/remotes/origin
master
tutorial $ cat .git/refs/remotes/origin/master
32606bc9ca5b1718720971410697b1d36bf13008
tutorial $


tutorial — bash — 80×24
bash
tutorial $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
tutorial $

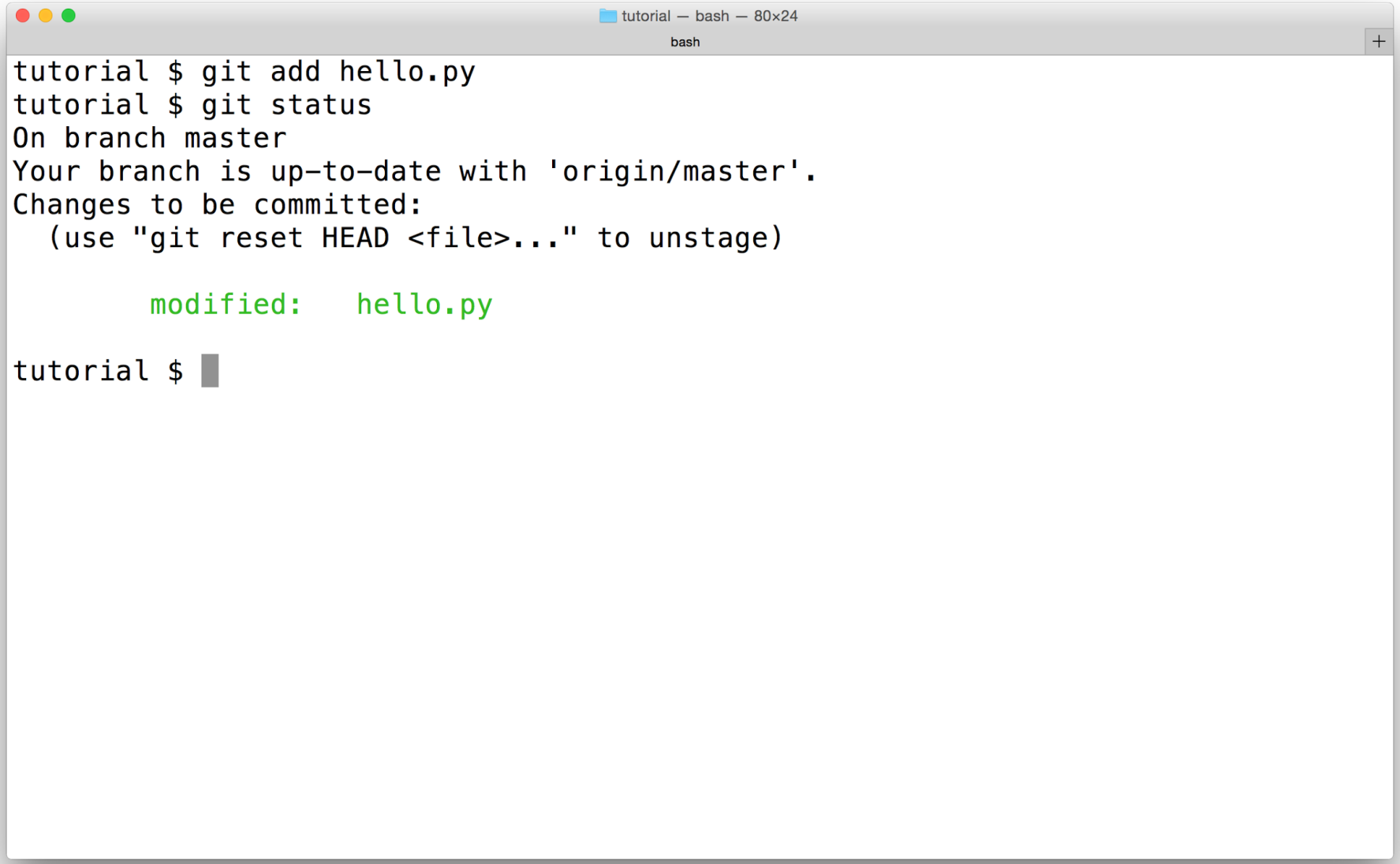


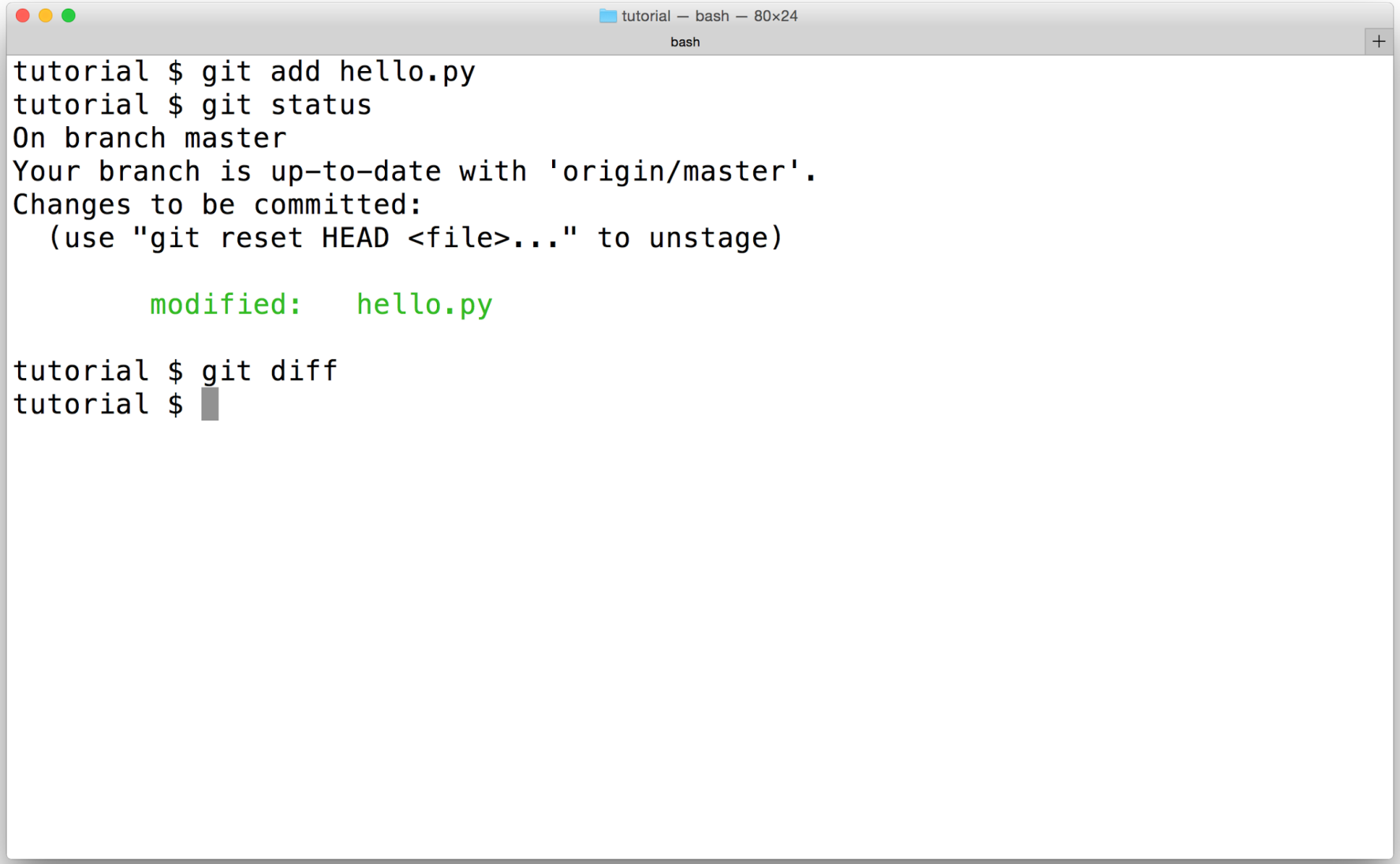

tutorial — bash — 80×24
bash
tutorial $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)
        modified: hello.py
no changes added to commit (use "git add" and/or "git commit -a")
tutorial $


tutorial — bash — 80×24
bash
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: hello.py
no changes added to commit (use "git add" and/or "git commit -a")
tutorial $ git diff
diff --git a/hello.py b/hello.py
index 4351743..1790a78 100644
--- a/hello.py
+++ b/hello.py
@@ -1 +1,9 @@
-print "Hello, world!"
+#!/usr/bin/env python
+
+
+def main ():
+ print "Hello, world!"
+
+
+if __name__ == "__main__":
+ main()
tutorial $
$ git diff [file]

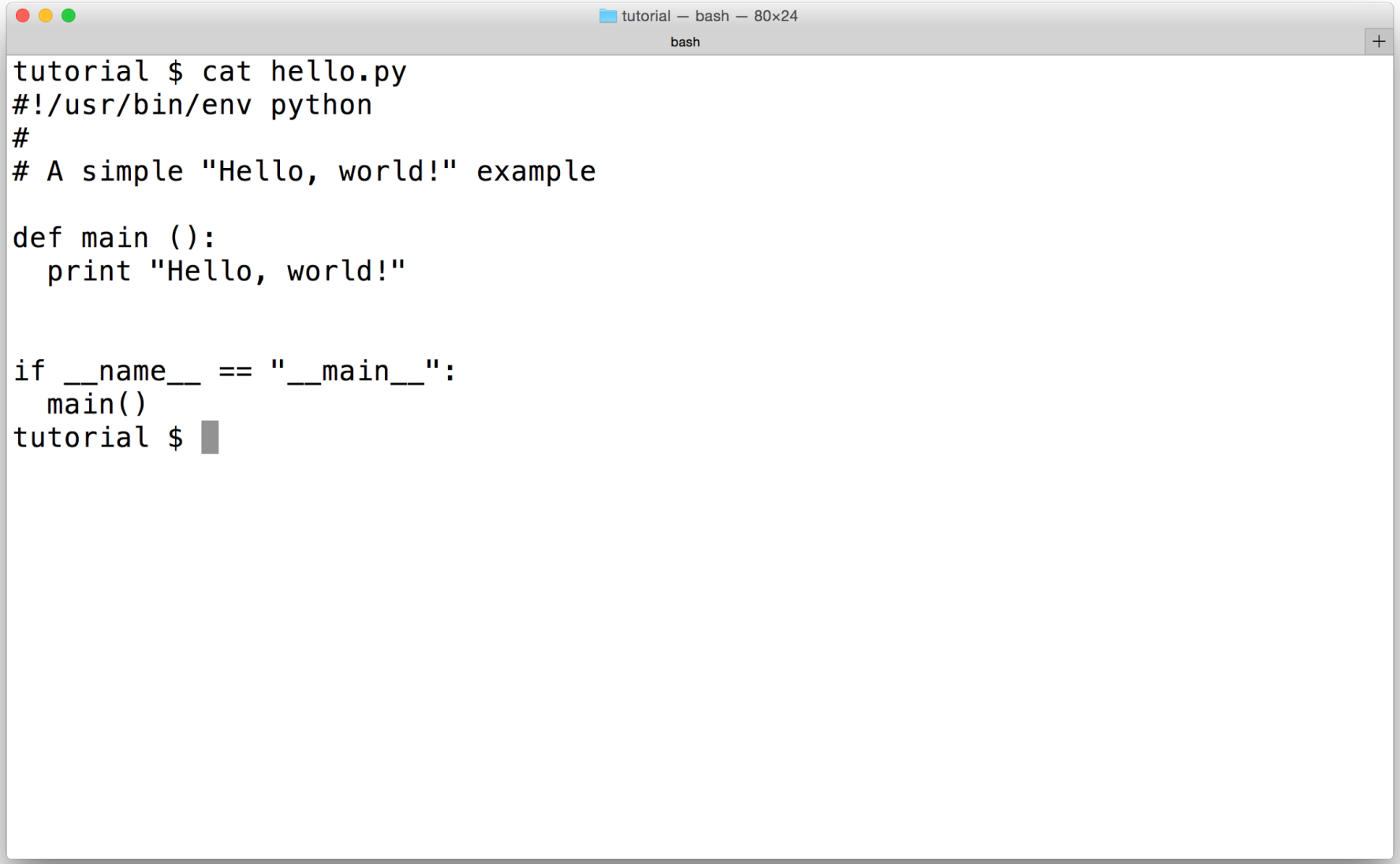


http://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository


tutorial — bash — 80×24
bash
tutorial $ git add hello.py
tutorial $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
        modified: hello.py
tutorial $

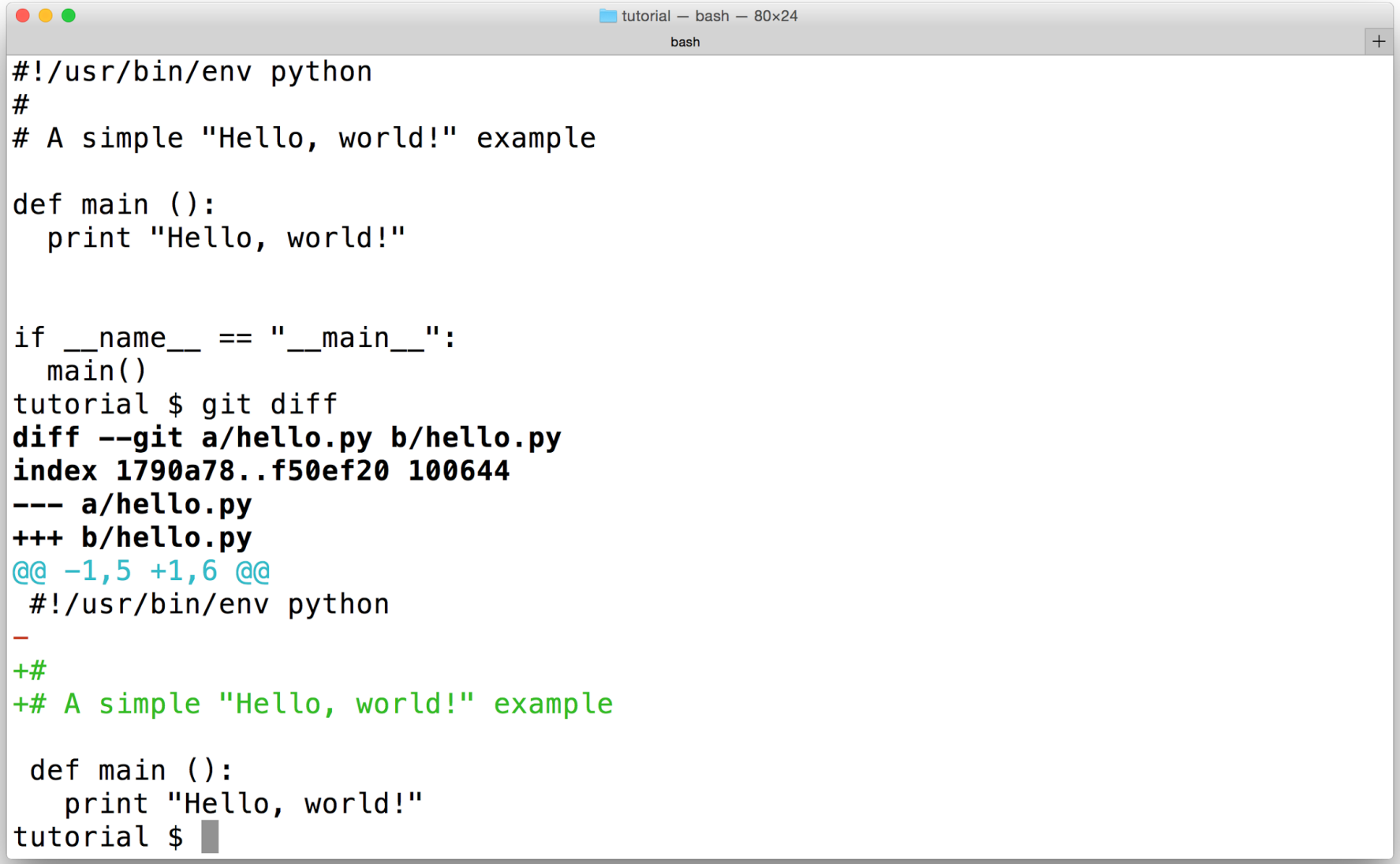

tutorial — bash — 80×24
bash
tutorial $ git add hello.py
tutorial $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
        modified: hello.py
tutorial $ git diff
tutorial $

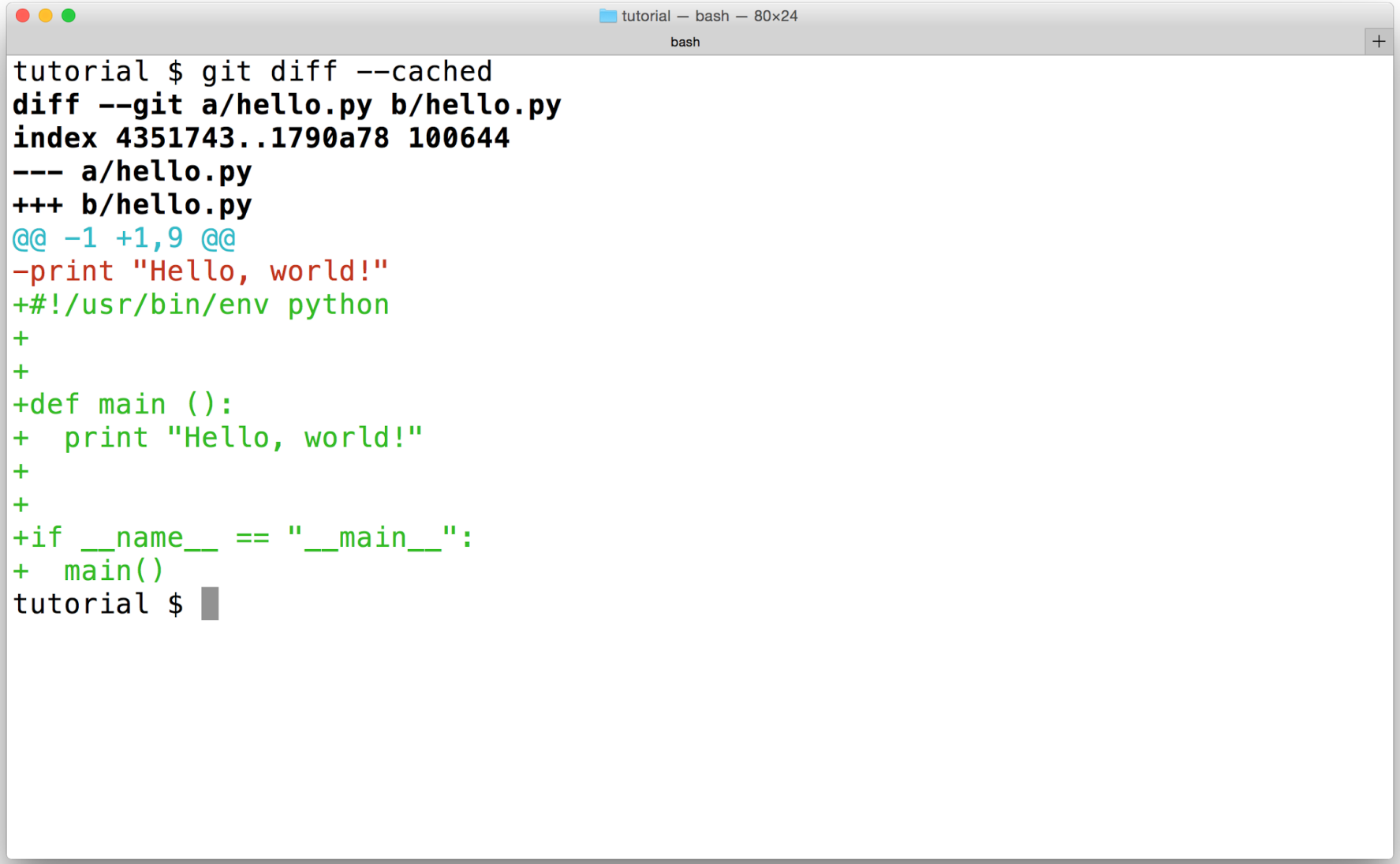

tutorial — bash — 80×24
bash
tutorial $ git diff --cached
diff --git a/hello.py b/hello.py
index 4351743..1790a78 100644
--- a/hello.py
+++ b/hello.py
@@ -1 +1,9 @@
-print "Hello, world!"
+#!/usr/bin/env python
+
+
+def main ():
+ print "Hello, world!"
+
+
+if __name__ == "__main__":
+ main()
tutorial $
$ git diff --cached [file]

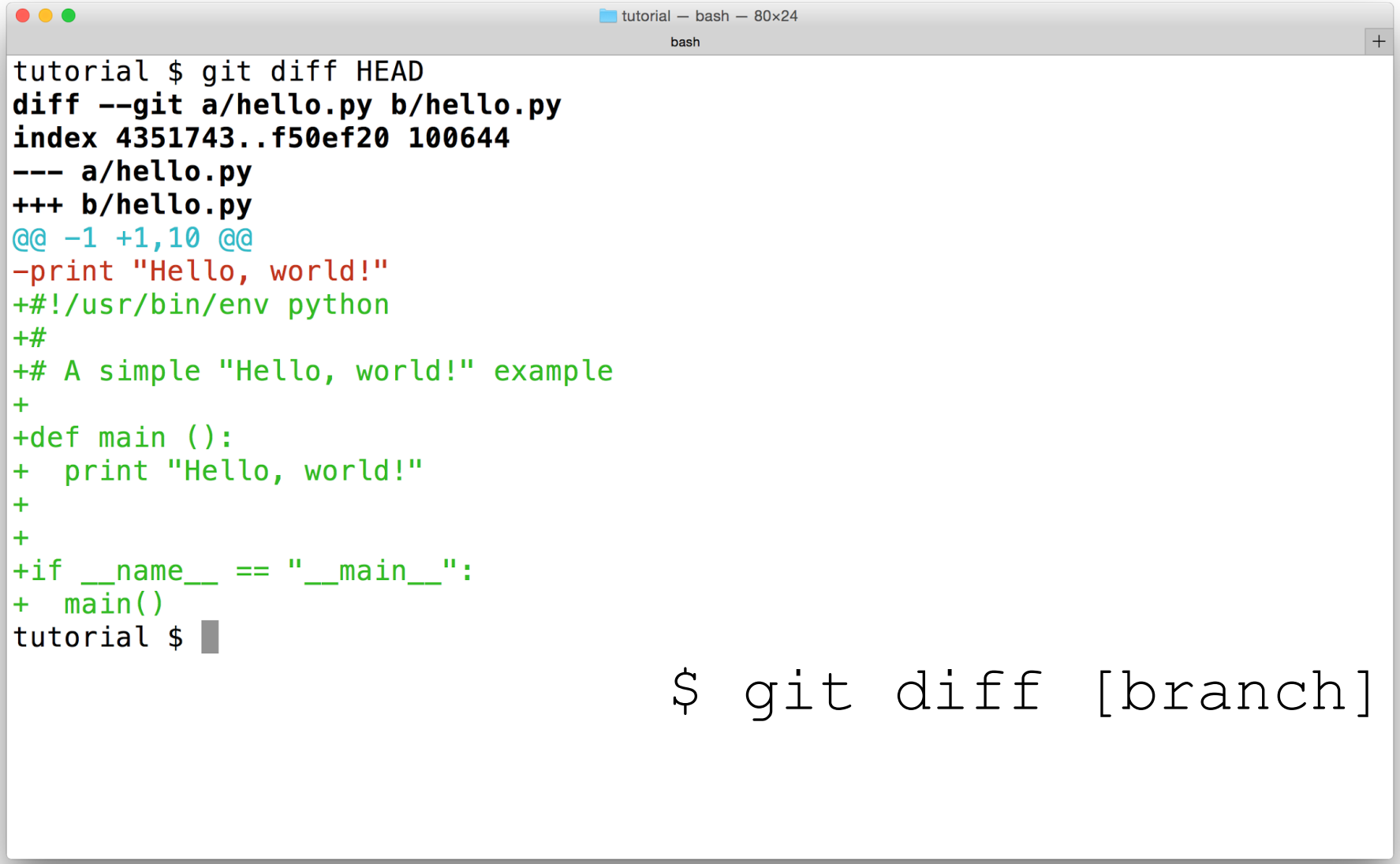

tutorial — bash — 80×24
bash
tutorial $ cat hello.py
#!/usr/bin/env python
#
A simple "Hello, world!" example

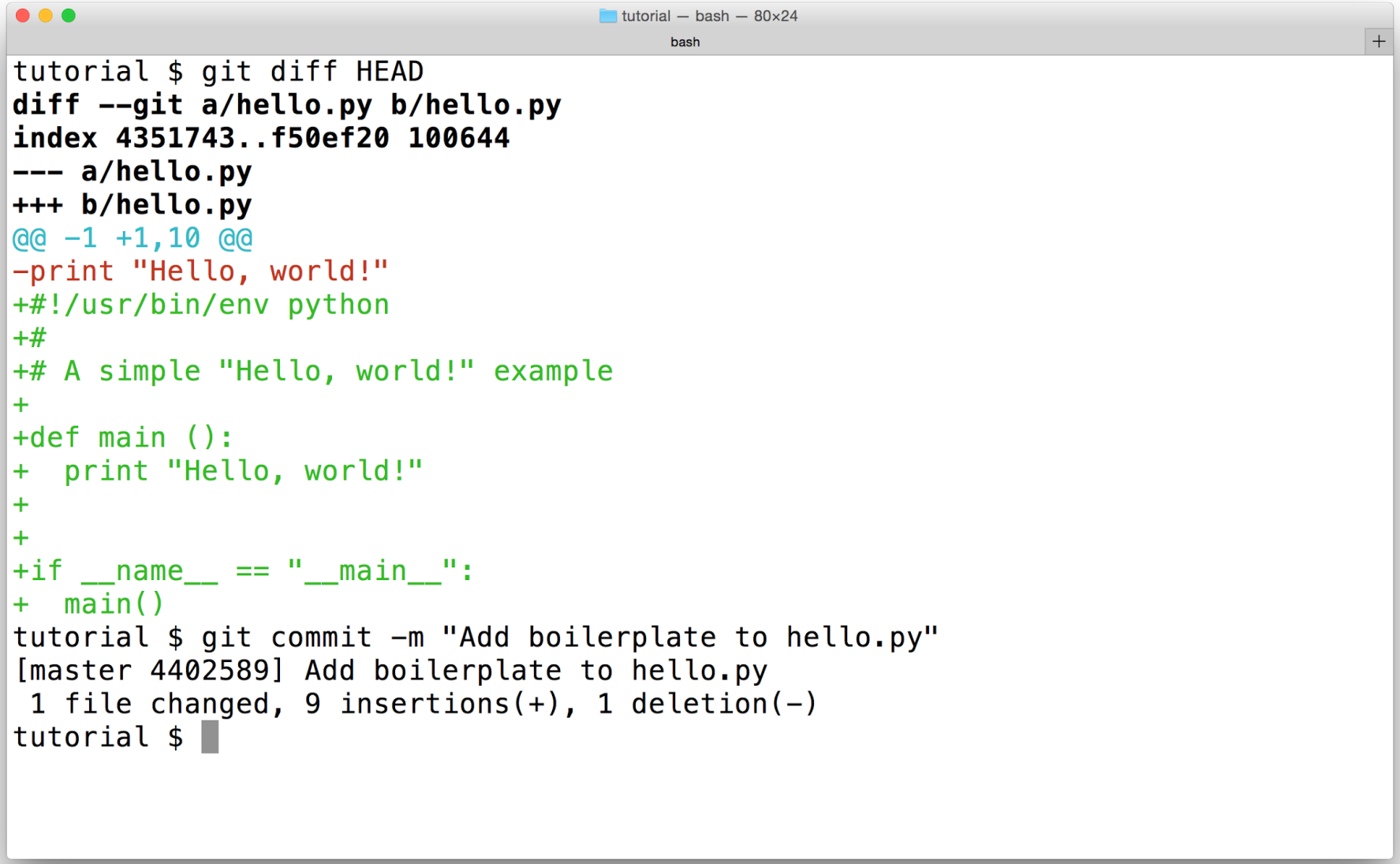
def main ():
  print "Hello, world!"

if __name__ == "__main__":
  main()
tutorial $

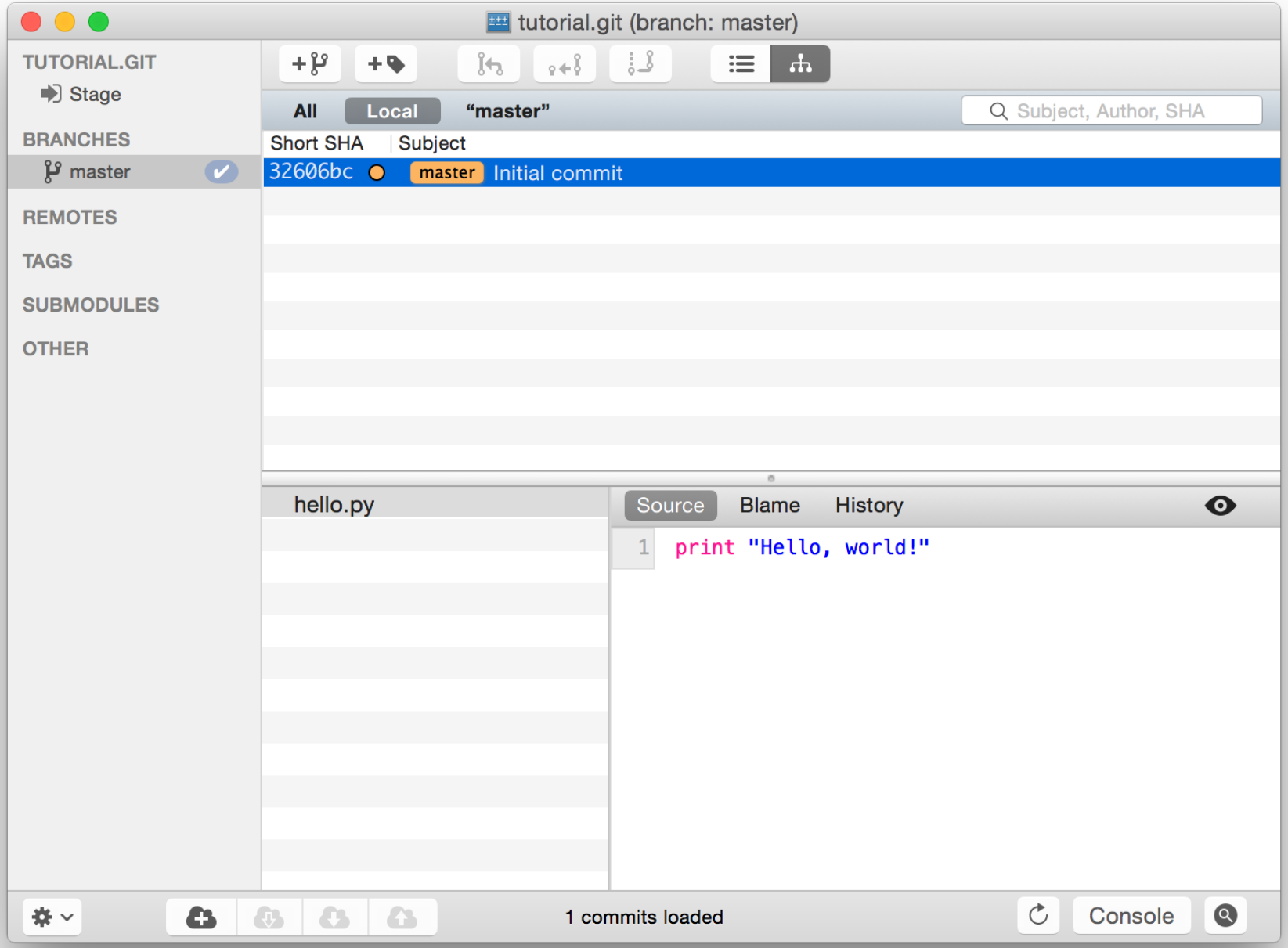

tutorial — bash — 80×24
bash
#!/usr/bin/env python
#
# A simple "Hello, world!" example
def main ():
print "Hello, world!"
if __name__ == "__main__":
main()
tutorial $ git diff
diff --git a/hello.py b/hello.py
index 1790a78..f50ef20 100644
--- a/hello.py
+++ b/hello.py
@@ -1,5 +1,6 @@
#!/usr/bin/env python
-
+#
+# A simple "Hello, world!" example
def main ():
print "Hello, world!"
tutorial $

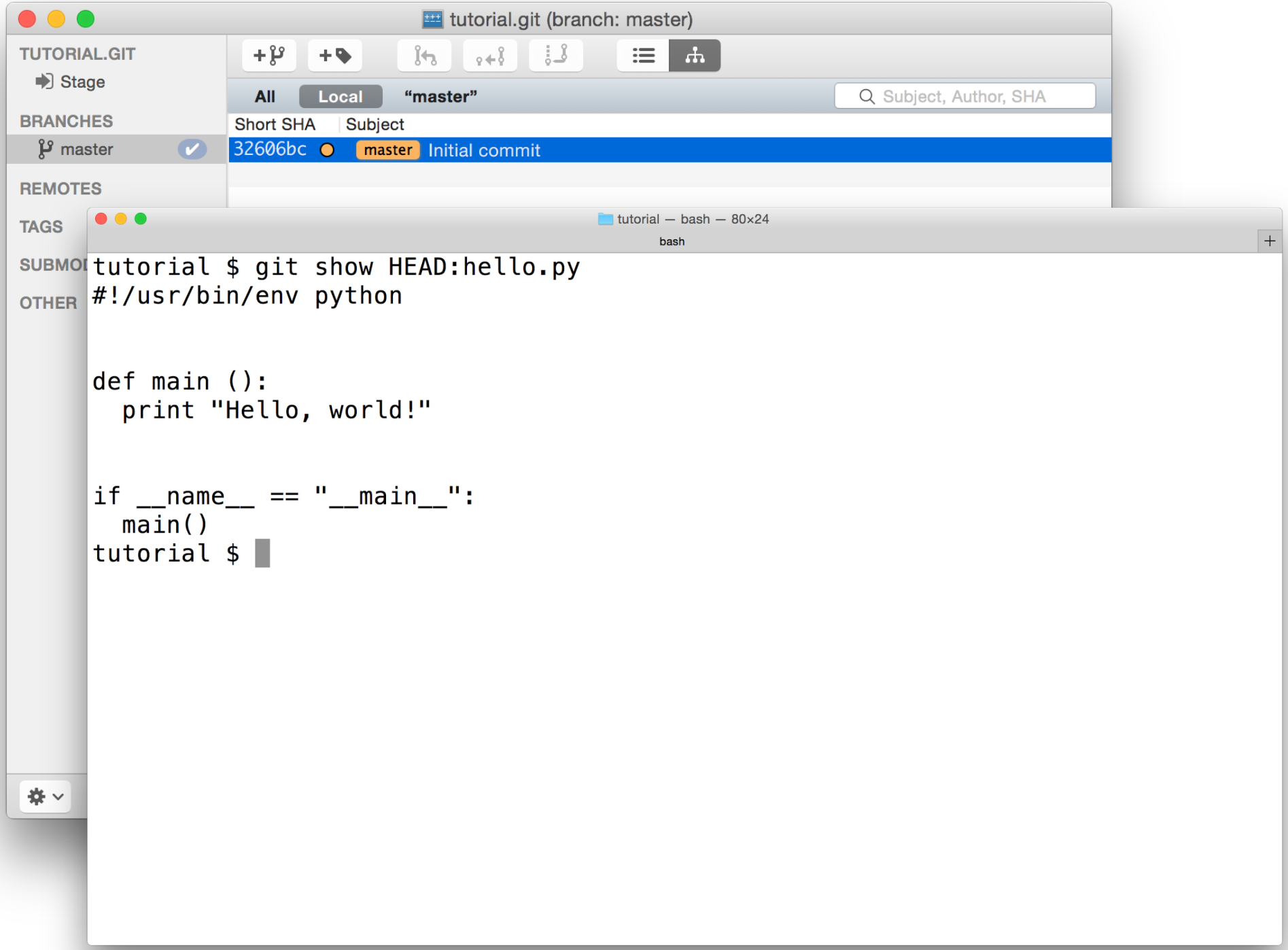

tutorial — bash — 80×24
bash
tutorial $ git diff --cached
diff --git a/hello.py b/hello.py
index 4351743..1790a78 100644
--- a/hello.py
+++ b/hello.py
@@ -1 +1,9 @@
-print "Hello, world!"
+#!/usr/bin/env python
+
+
+def main ():
+ print "Hello, world!"
+
+
+if __name__ == "__main__":
+ main()
tutorial $

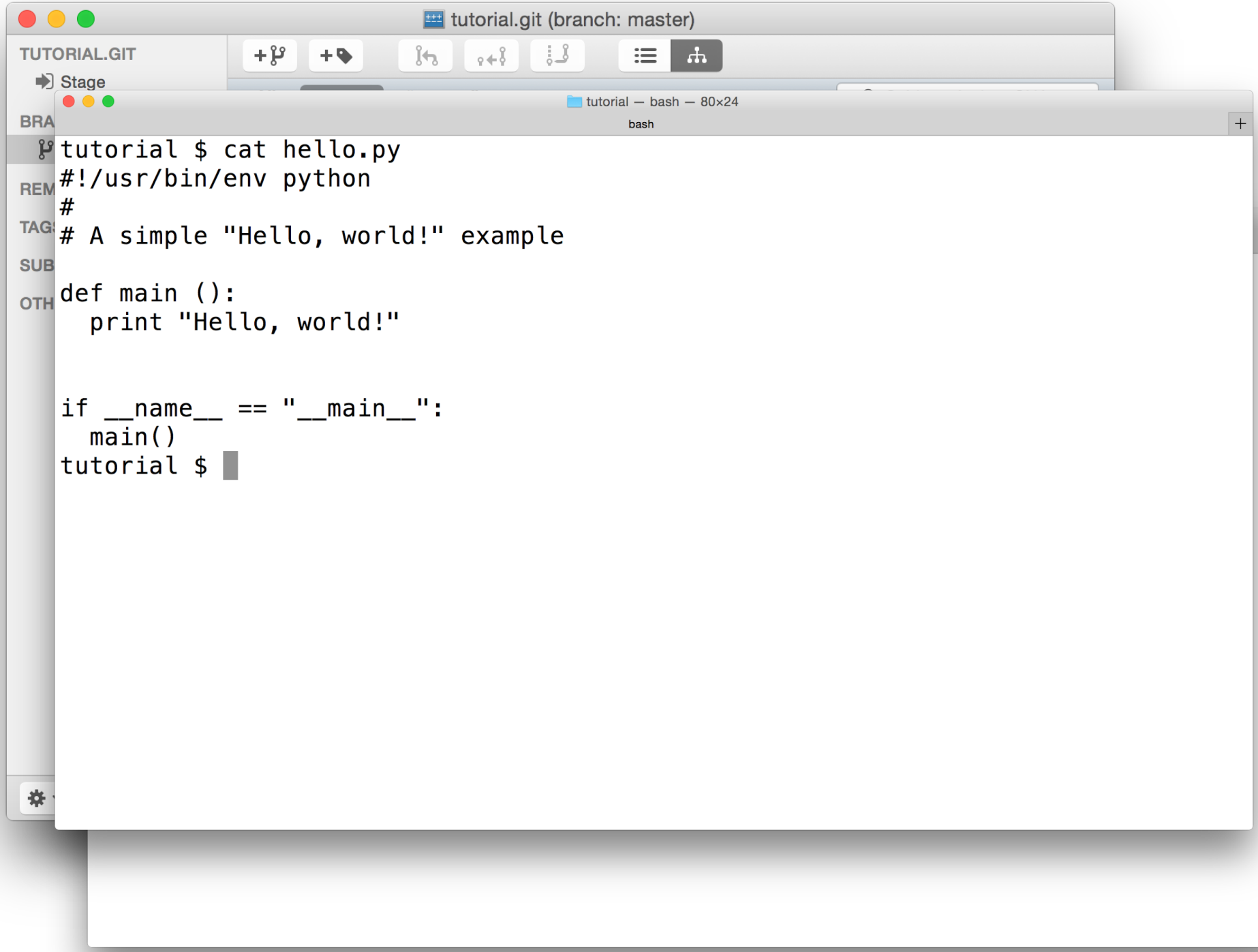

tutorial — bash — 80×24
bash
tutorial $ git diff HEAD
diff --git a/hello.py b/hello.py
index 4351743..f50ef20 100644
--- a/hello.py
+++ b/hello.py
@@ -1 +1,10 @@
-print "Hello, world!"
+#!/usr/bin/env python
+#
+# A simple "Hello, world!" example
+
+def main ():
+ print "Hello, world!"
+
+
+if __name__ == "__main__":
+ main()
tutorial $
$ git diff [branch]

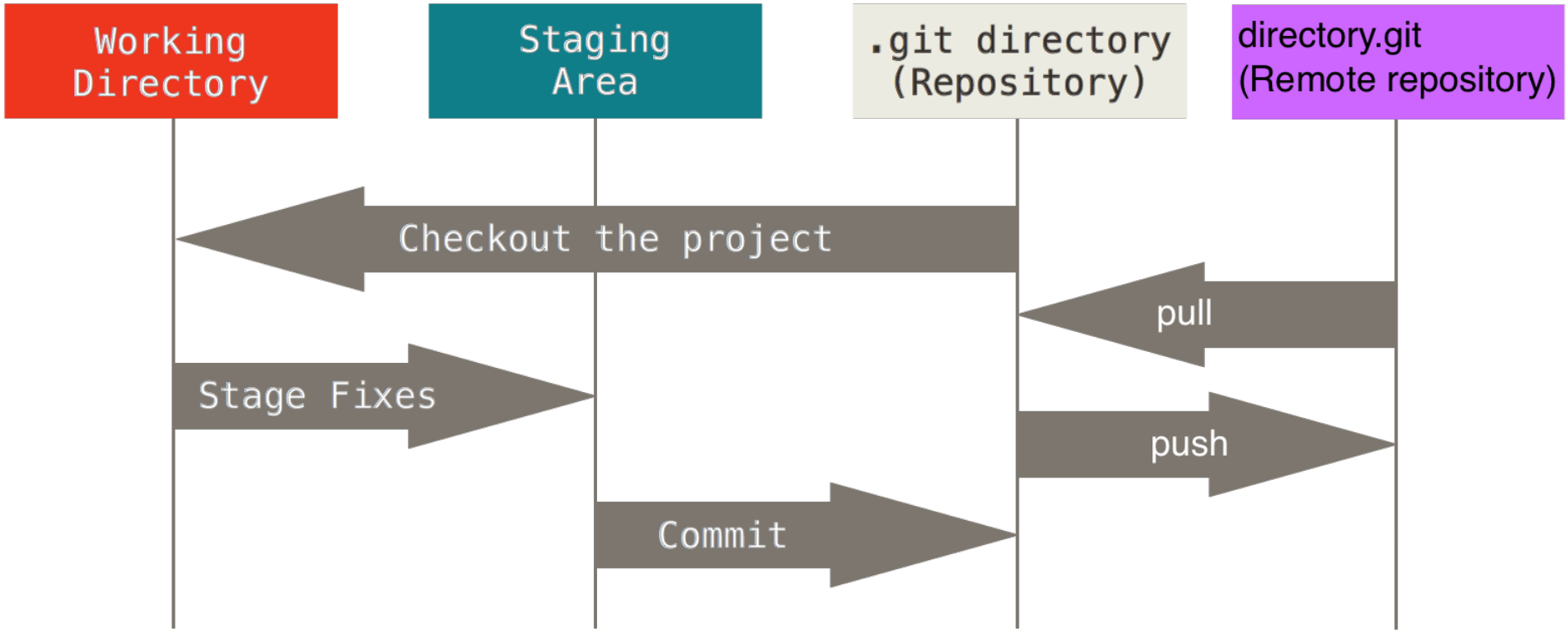

tutorial — bash — 80×24
bash
tutorial $ git diff HEAD
diff --git a/hello.py b/hello.py
index 4351743..f50ef20 100644
--- a/hello.py
+++ b/hello.py
@@ -1 +1,10 @@
-print "Hello, world!"
+#!/usr/bin/env python
+#
+# A simple "Hello, world!" example
+
+def main ():
+ print "Hello, world!"
+
+
+if __name__ == "__main__":
+ main()
tutorial $ git commit -m "Add boilerplate to hello.py"
[master 4402589] Add boilerplate to hello.py
 1 file changed, 9 insertions(+), 1 deletion(-)
tutorial $

Stop!

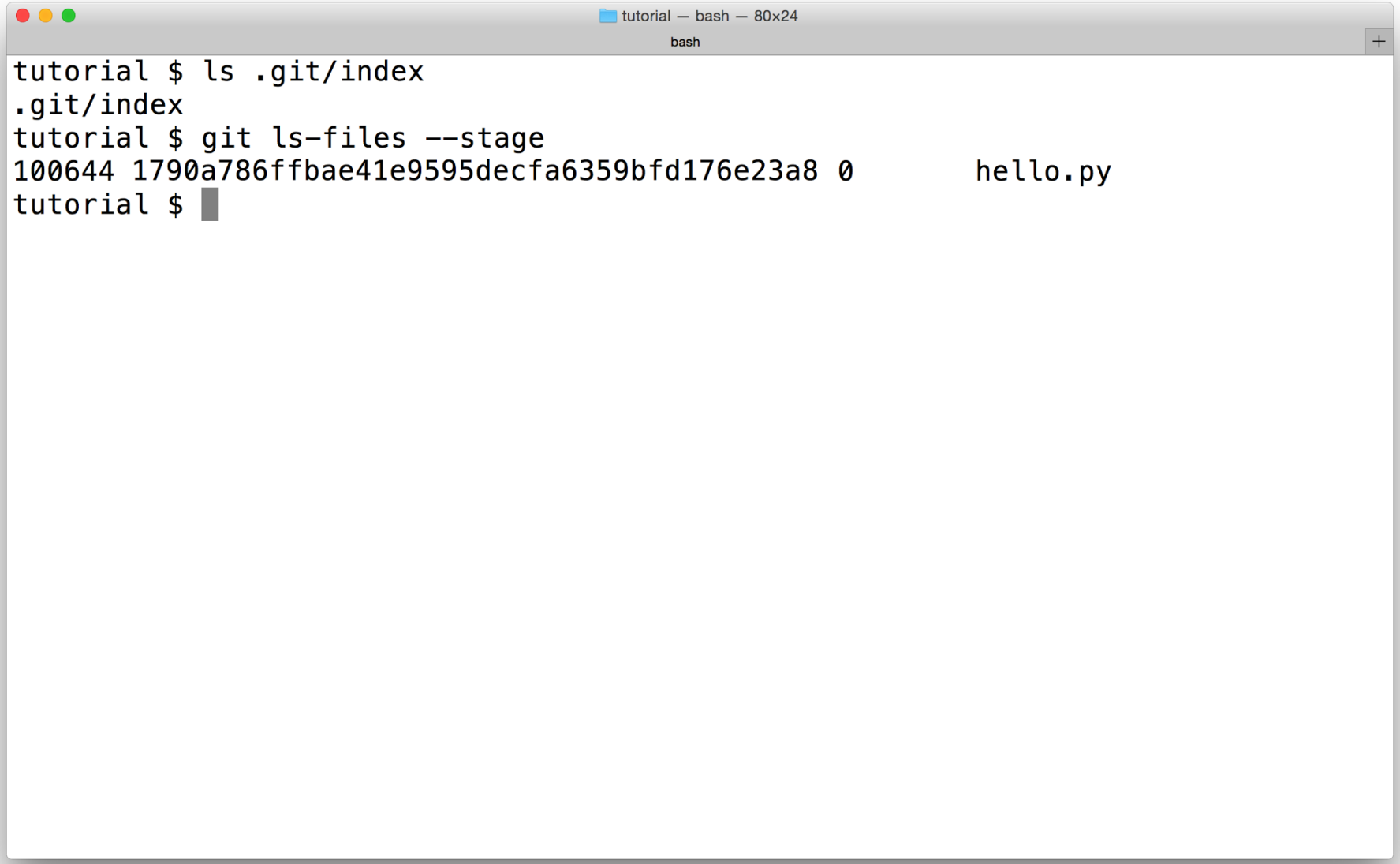

tutorial.git (branch: master)
TUTORIAL.GIT
Stage
BRANCHES
master
REMOTES
TAGS
SUBMODULES
OTHER
All
Local
“master”
Subject, Author, SHA
Short SHA
Subject
32606bc
master
Initial commit
hello.py
Source
Blame
History
1
print "Hello, world!"
1 commits loaded
Console


tutorial.git (branch: master)
TUTORIAL.GIT
Stage
BRANCHES
master
REMOTES
TAGS
SUBMOD
OTHER
All
Local
"master"
Subject, Author, SHA
Short SHA
Subject
32606bc
master
Initial commit
tutorial — bash — 80×24
bash
tutorial $ git show HEAD:hello.py
#!/usr/bin/env python
def main ():
print "Hello, world!"
if __name__ == "__main__":
main()
tutorial $

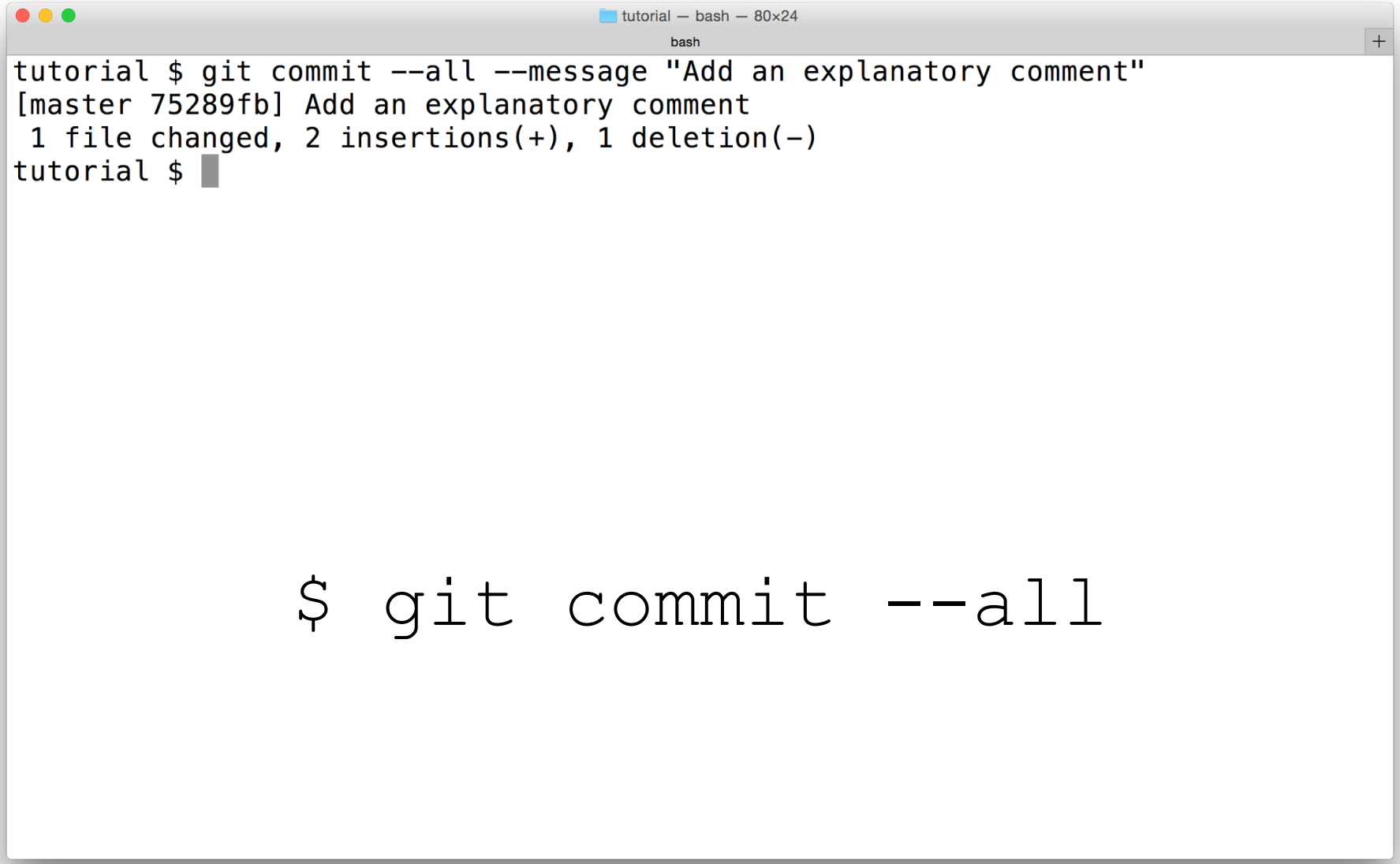

tutorial.git (branch: master)
TUTORIAL.GIT
Stage
tutorial — bash — 80×24
bash
tutorial $ cat hello.py
#!/usr/bin/env python
#
# A simple "Hello, world!" example
def main ():
print "Hello, world!"
if __name__ == "__main__":
main()
tutorial $

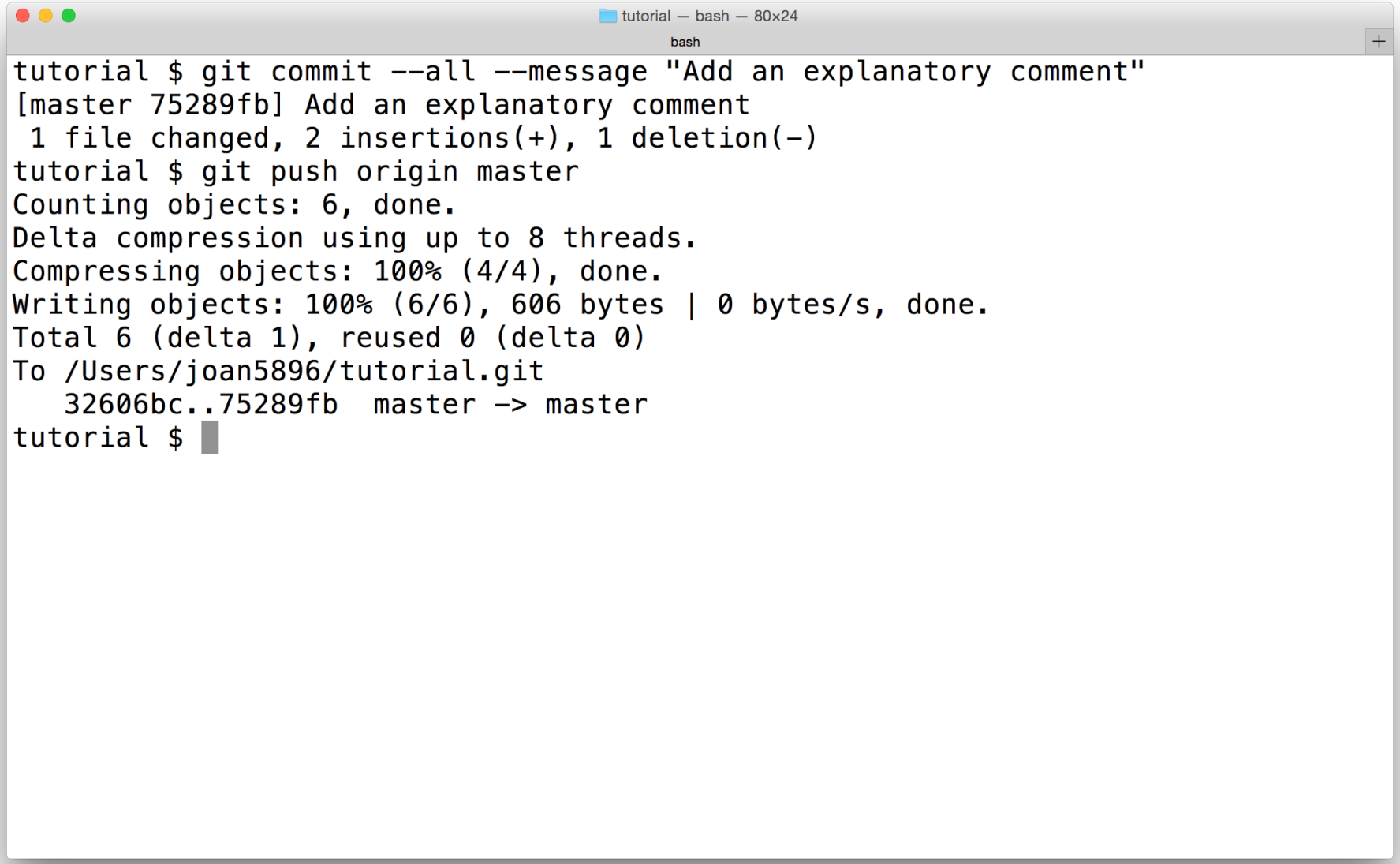


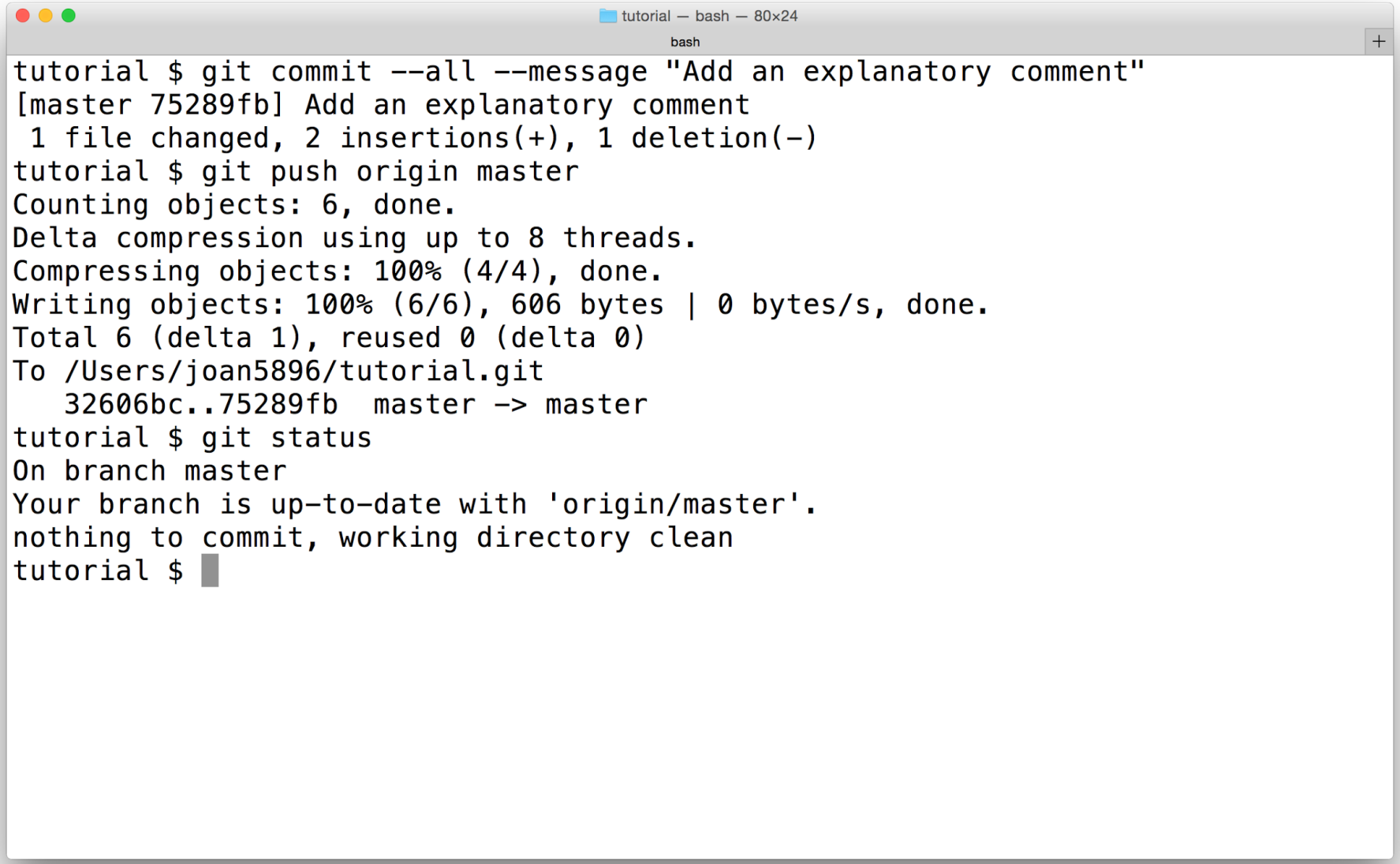
http://git-scm.com/book/en/v2/Getting-Started-Git-Basics

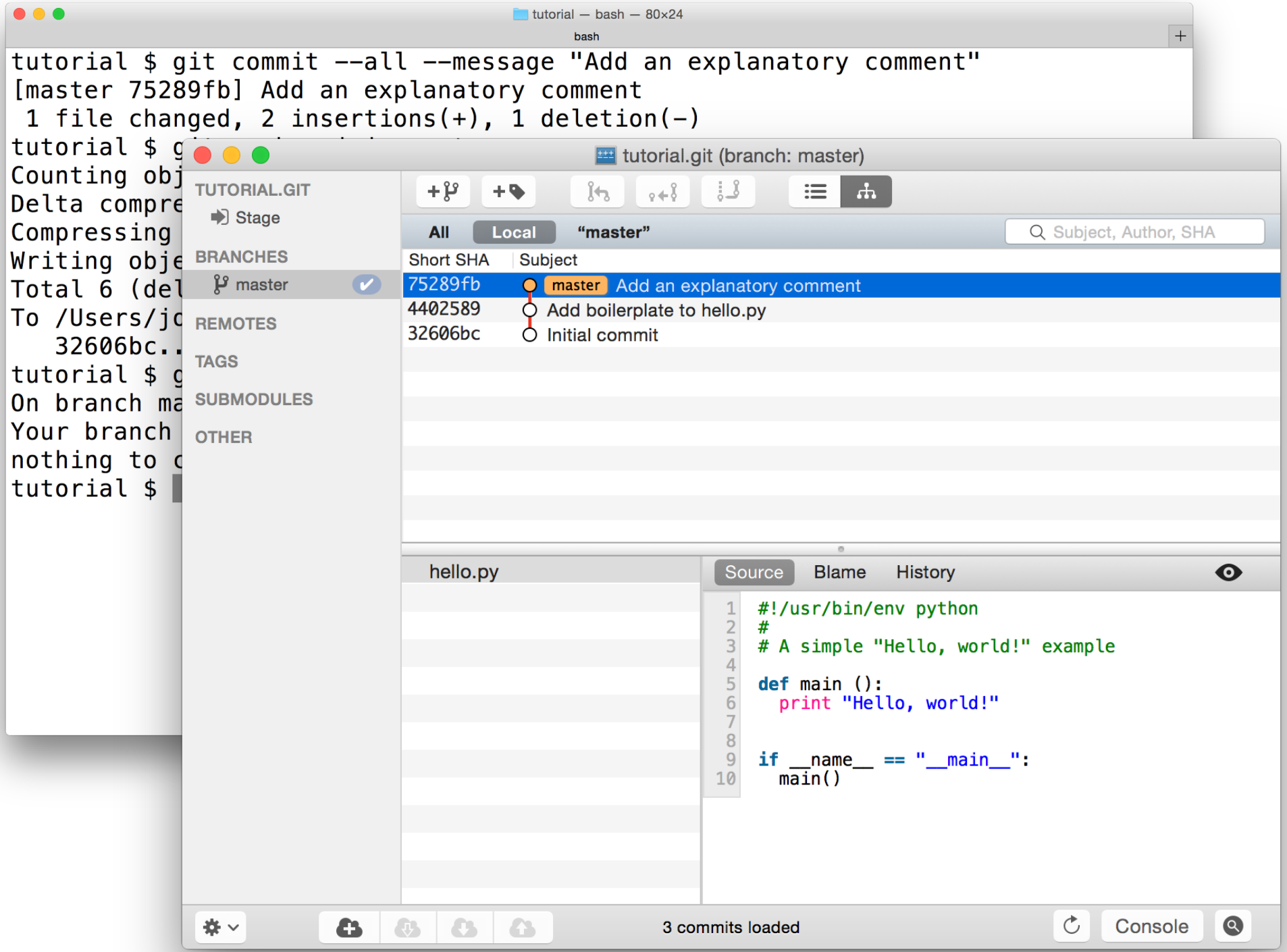

tutorial — bash — 80×24
bash
tutorial $ ls .git/index
.git/index
tutorial $ git ls-files --stage
100644 1790a786ffbae41e9595decfa6359bfd176e23a8 0 hello.py
tutorial $

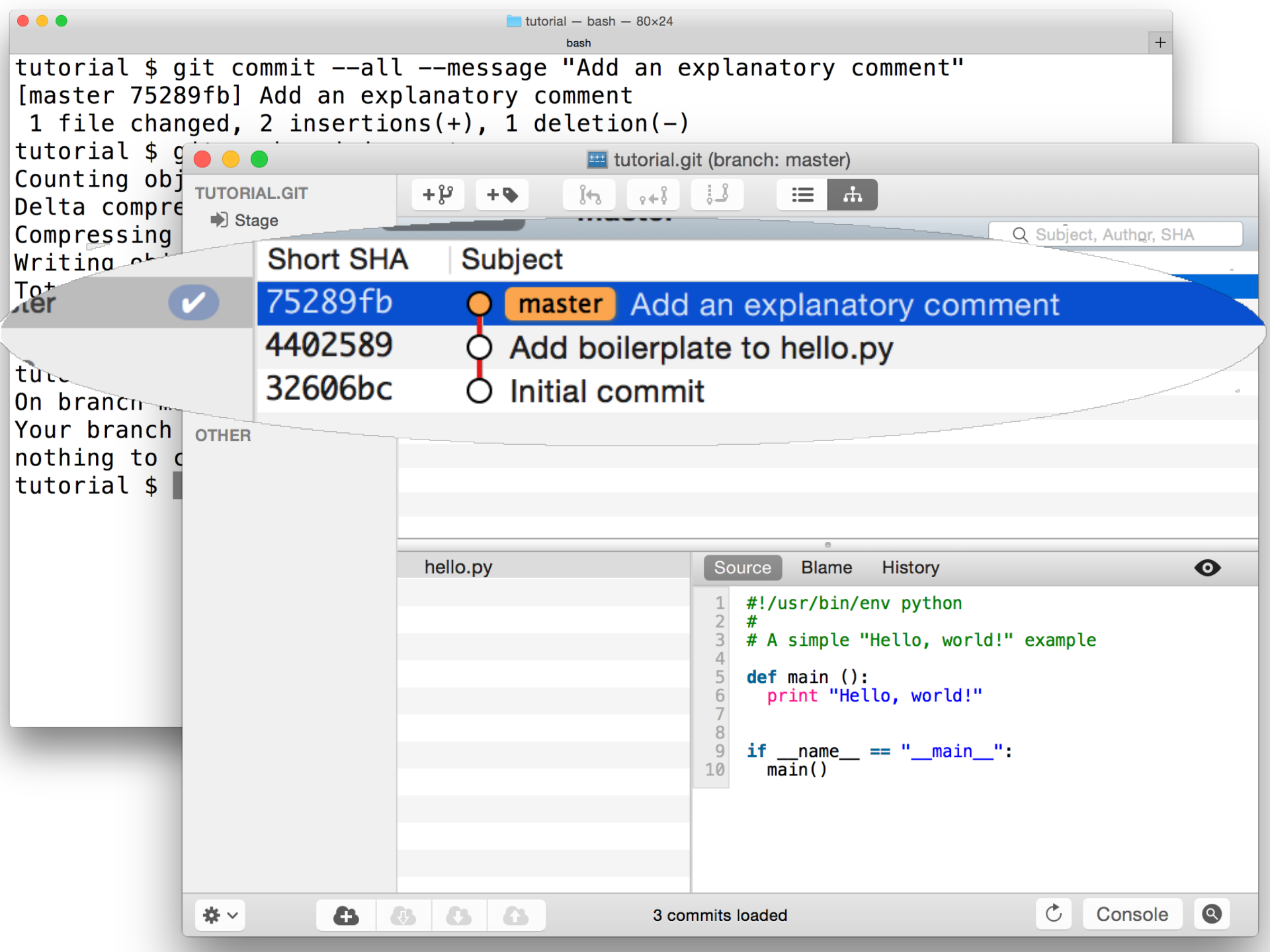

tutorial — bash — 80×24
bash
tutorial $ git commit hello.py
$ git commit [file]


tutorial — bash — 80×24
bash
tutorial $ git commit --all --message "Add an explanatory comment"
[master 75289fb] Add an explanatory comment
 1 file changed, 2 insertions(+), 1 deletion(-)
tutorial $
$ git commit --all

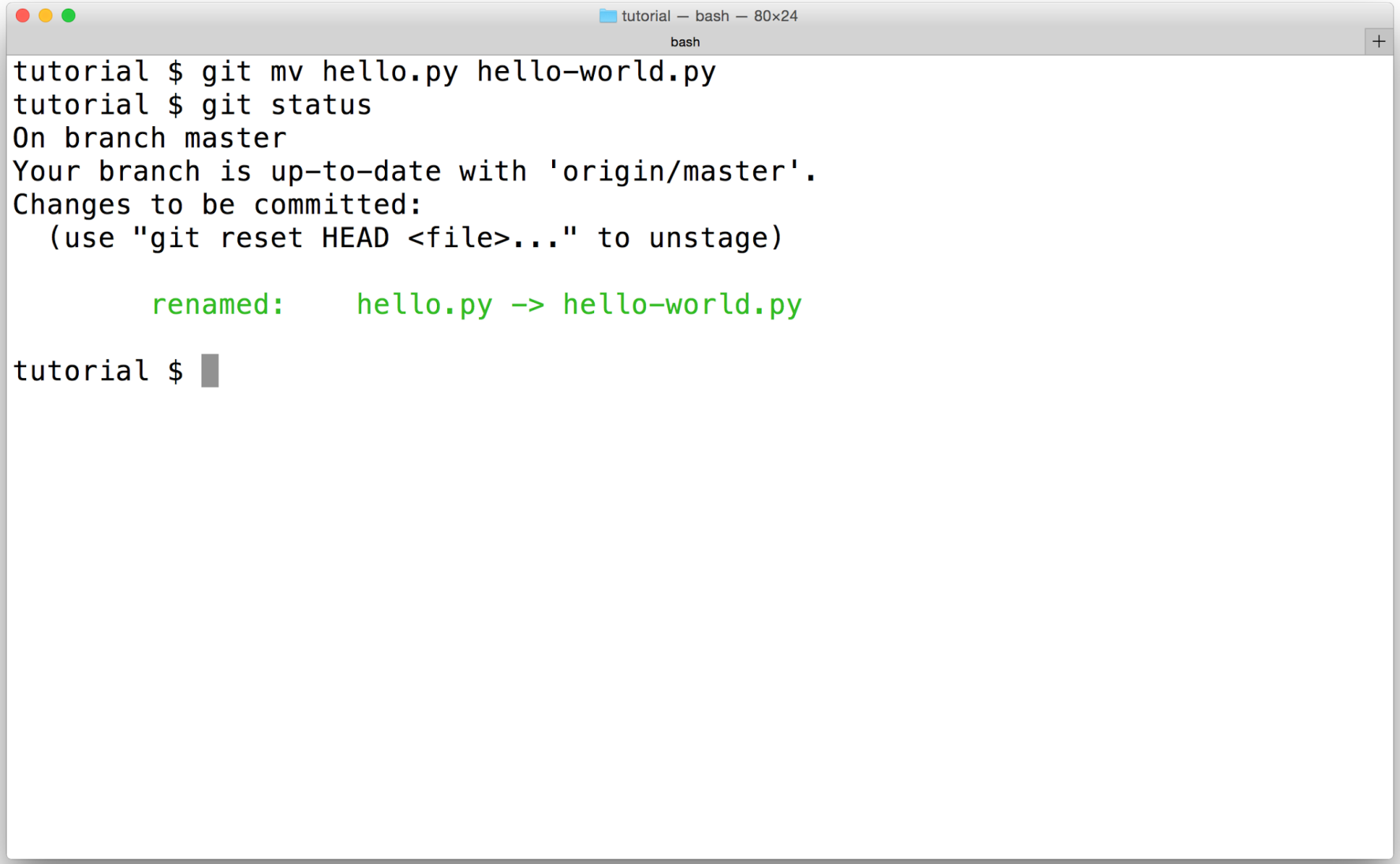

tutorial — bash — 80×24
bash
tutorial $ git commit --all --message "Add an explanatory comment"
[master 75289fb] Add an explanatory comment
 1 file changed, 2 insertions(+), 1 deletion(-)
tutorial $ git push origin master
Counting objects: 6, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 606 bytes | 0 bytes/s, done.
Total 6 (delta 1), reused 0 (delta 0)
To /Users/joan5896/tutorial.git
   32606bc..75289fb master -> master
tutorial $

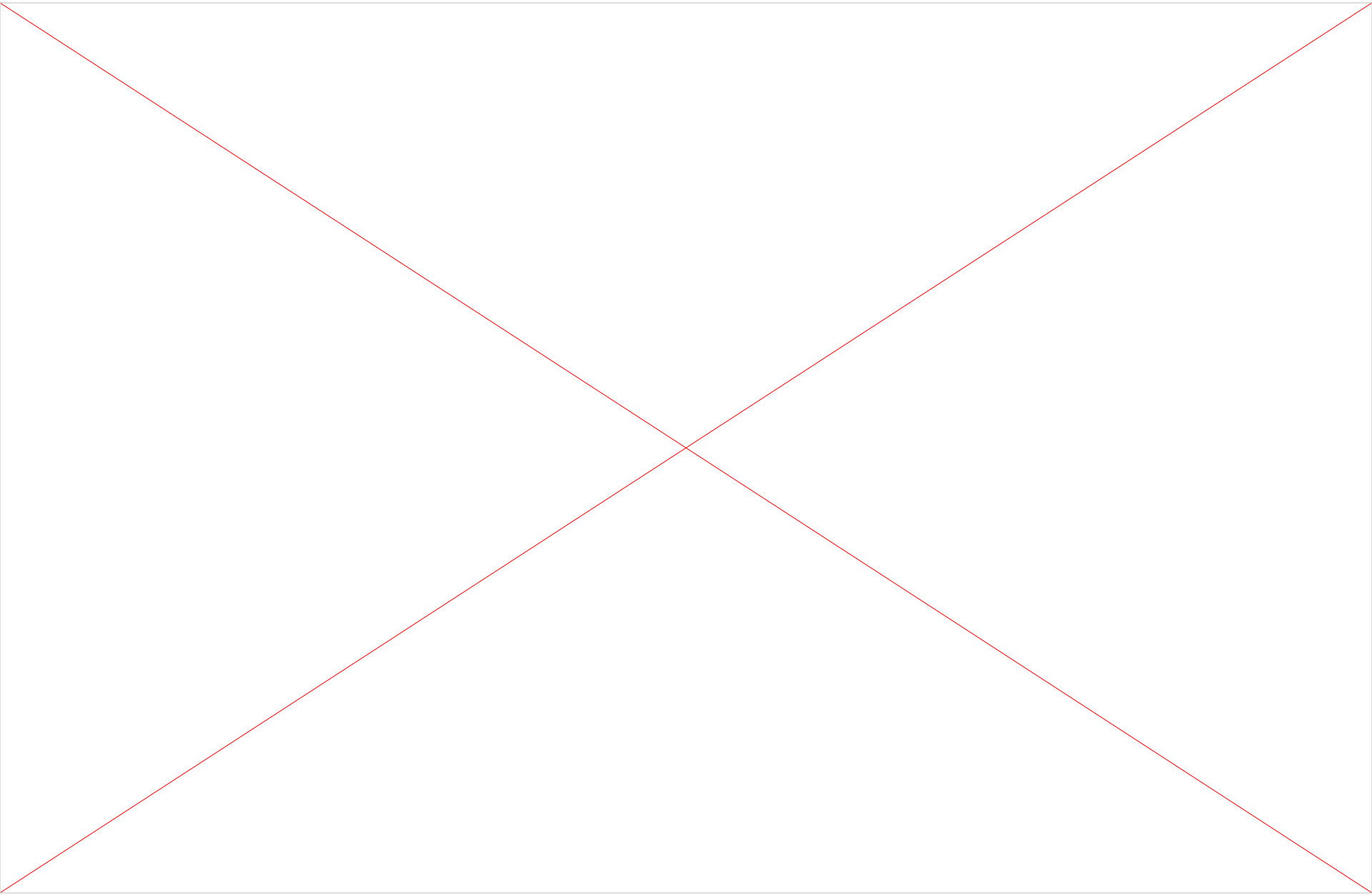

tutorial — bash — 80×24
bash
tutorial $ git commit --all --message "Add an explanatory comment"
[master 75289fb] Add an explanatory comment
 1 file changed, 2 insertions(+), 1 deletion(-)
tutorial $ git push origin master
Counting objects: 6, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 606 bytes | 0 bytes/s, done.
Total 6 (delta 1), reused 0 (delta 0)
To /Users/joan5896/tutorial.git
   32606bc..75289fb master -> master
tutorial $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
tutorial $


tutorial — bash — 80×24
bash
tutorial $ git commit --all --message "Add an explanatory comment"
[master 75289fb] Add an explanatory comment
1 file changed, 2 insertions(+), 1 deletion(-)
tutorial $ g
Counting obj
Delta compre
Compressing
Writing obje
Total 6 (del
To /Users/jc
32606bc..
tutorial $ g
On branch ma
Your branch
nothing to c
tutorial $
tutorial.git (branch: master)
TUTORIAL.GIT
Stage
BRANCHES
master
REMOTES
TAGS
SUBMODULES
OTHER
All
Local
"master"
Subject, Author, SHA
Short SHA
Subject
75289fb
master
Add an explanatory comment
4402589
Add boilerplate to hello.py
32606bc
Initial commit
hello.py
Source
Blame
History
#!/usr/bin/env python
#
# A simple "Hello, world!" example
def main ():
print "Hello, world!"
if __name__ == "__main__":
main()
3 commits loaded
Console

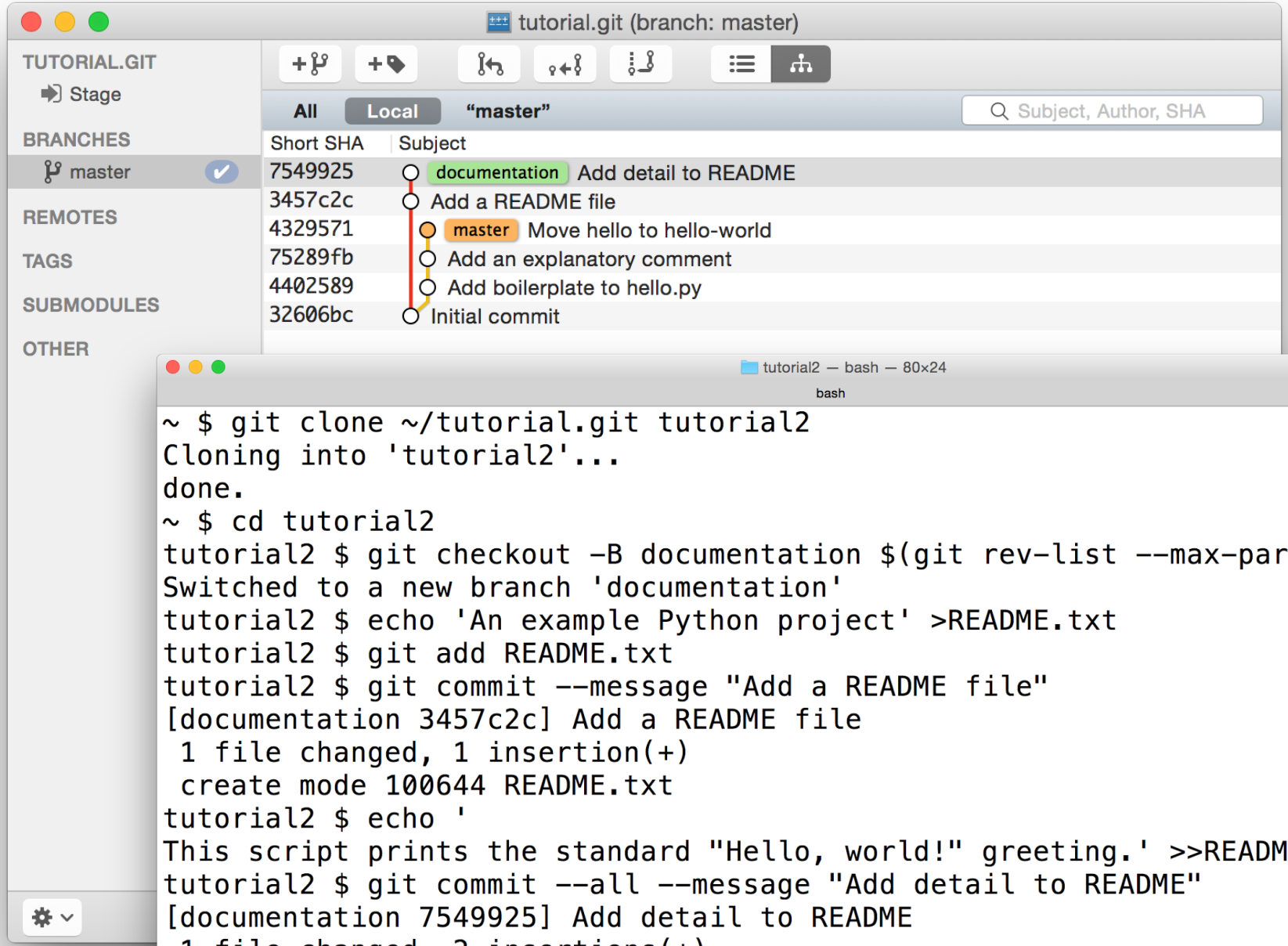

tutorial — bash — 80×24
bash
tutorial $ git commit --all --message "Add an explanatory comment"
[master 75289fb] Add an explanatory comment
1 file changed, 2 insertions(+), 1 deletion(-)
tutorial $
Counting obj
Delta compre
Compressing
Writing
To
On branch
Your branch
nothing to c
tutorial $
tutorial.git (branch: master)
TUTORIAL.GIT
Stage
OTHER
Subject, Author, SHA
Short SHA
Subject
75289fb
master
Add an explanatory comment
4402589
Add boilerplate to hello.py
32606bc
Initial commit
hello.py
Source
Blame
History
#!/usr/bin/env python
#
# A simple "Hello, world!" example
def main ():
print "Hello, world!"
if __name__ == "__main__":
main()
3 commits loaded
Console


tutorial — bash — 80×24
bash
tutorial $ git mv hello.py hello-world.py
tutorial $
$ git mv <src> <dest>


tutorial — bash — 80×24
bash
tutorial $ git mv hello.py hello-world.py
tutorial $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
        renamed: hello.py -> hello-world.py
tutorial $



Branching and merging



tutorial2 — bash — 80×24

bash

```
~ $ git clone ~/tutorial.git tutorial2
Cloning into 'tutorial2'...
done.
~ $ cd tutorial2
tutorial2 $ git checkout -B documentation $(git rev-list --max-parents=0 HEAD)
Switched to a new branch 'documentation'
tutorial2 $ echo 'An example Python project' >README.txt
tutorial2 $ git add README.txt
tutorial2 $ git commit --message "Add a README file"
[documentation 3457c2c] Add a README file
 1 file changed, 1 insertion(+)
 create mode 100644 README.txt
tutorial2 $ echo '
This script prints the standard "Hello, world!" greeting.' >>README.txt
tutorial2 $ git commit --all --message "Add detail to README"
[documentation 7549925] Add detail to README
 1 file changed, 2 insertions(+)
tutorial2 $
```


tutorial — bash — 80×24
bash
tutorial $ git fetch origin
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
From /Users/joan5896/tutorial
 * [new branch]   documentation -> origin/documentation
tutorial $
$ git fetch <remote>

Fetch:
To copy commits from the remote repository to the local repository

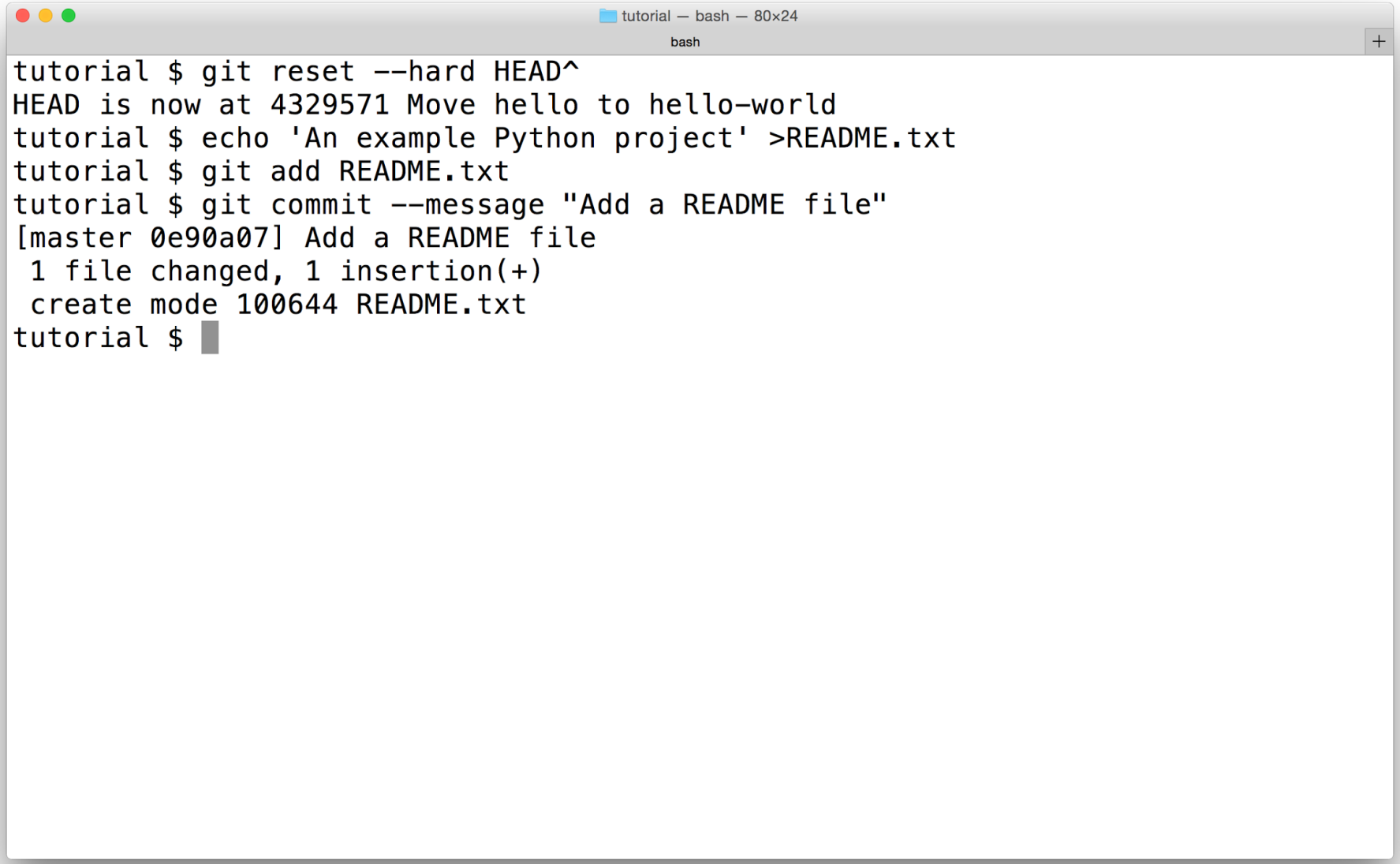

tutorial — bash — 80×24
bash
tutorial $ git merge origin/documentation
Merge made by the 'recursive' strategy.
README.txt | 3 +++
1 file changed, 3 insertions(+)
create mode 100644 README.txt
tutorial $
$ git merge <branch>

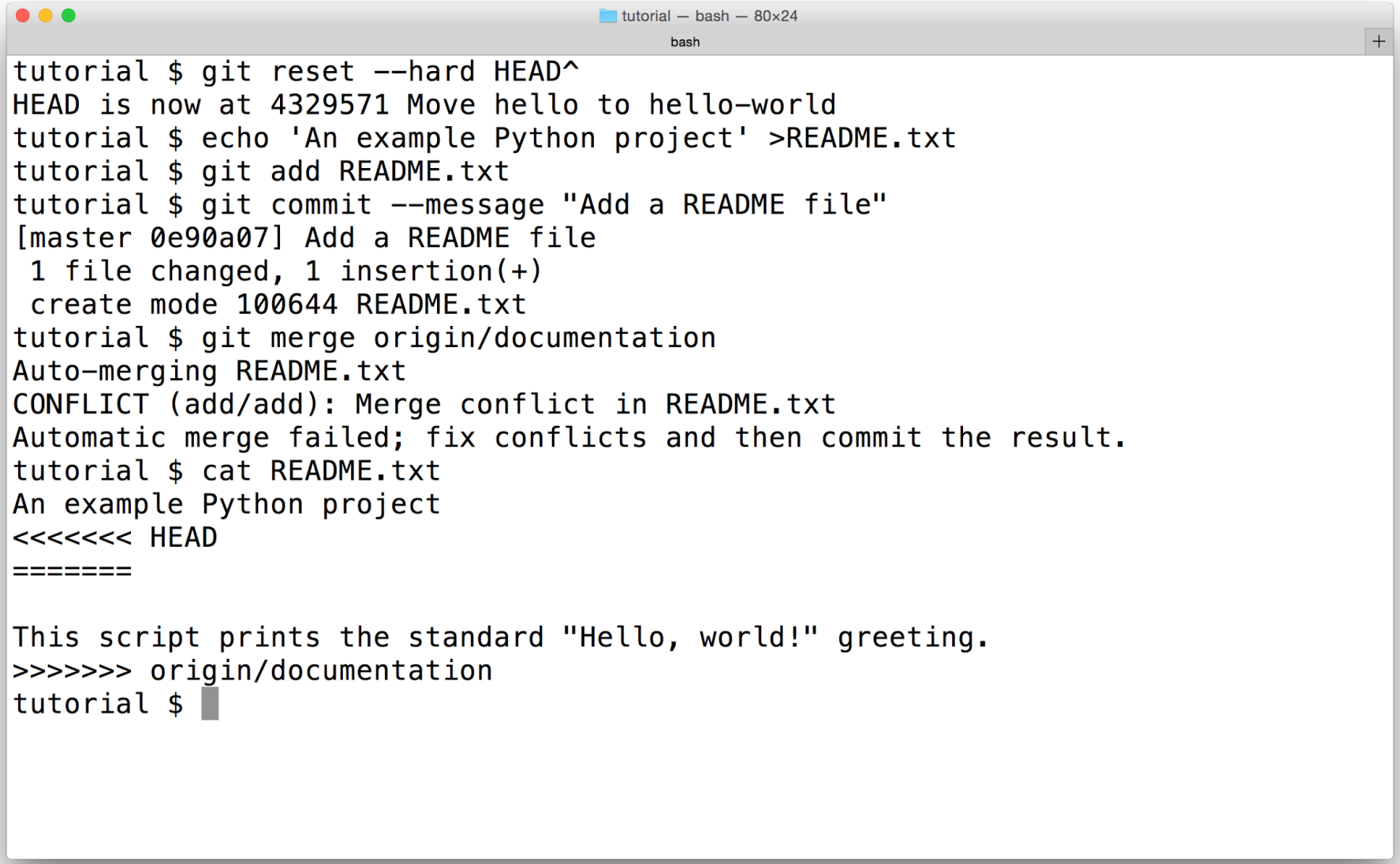
Merge:
To combine the content from two (or more) branches into a single branch

```
$ git pull <branch>
```

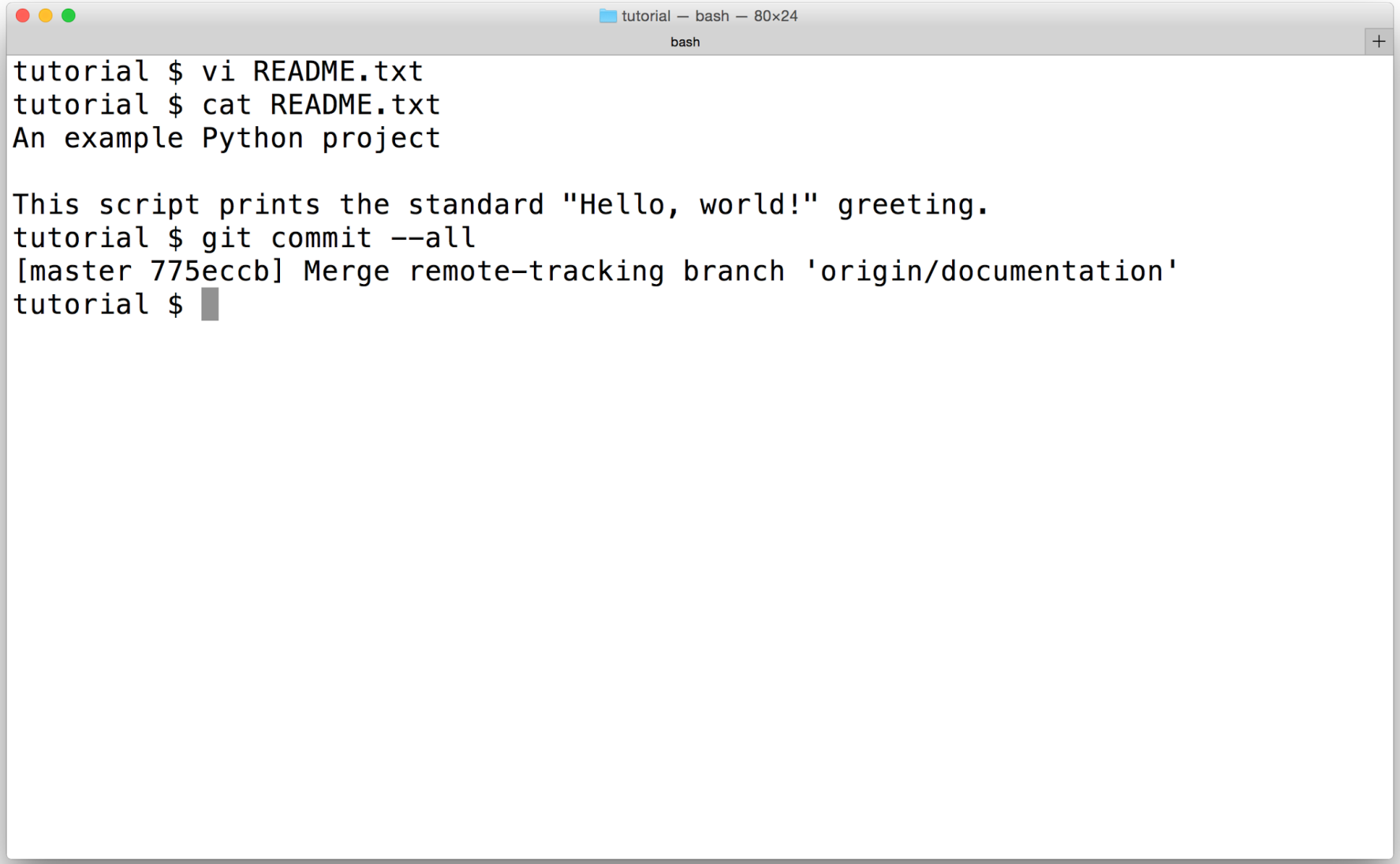
==

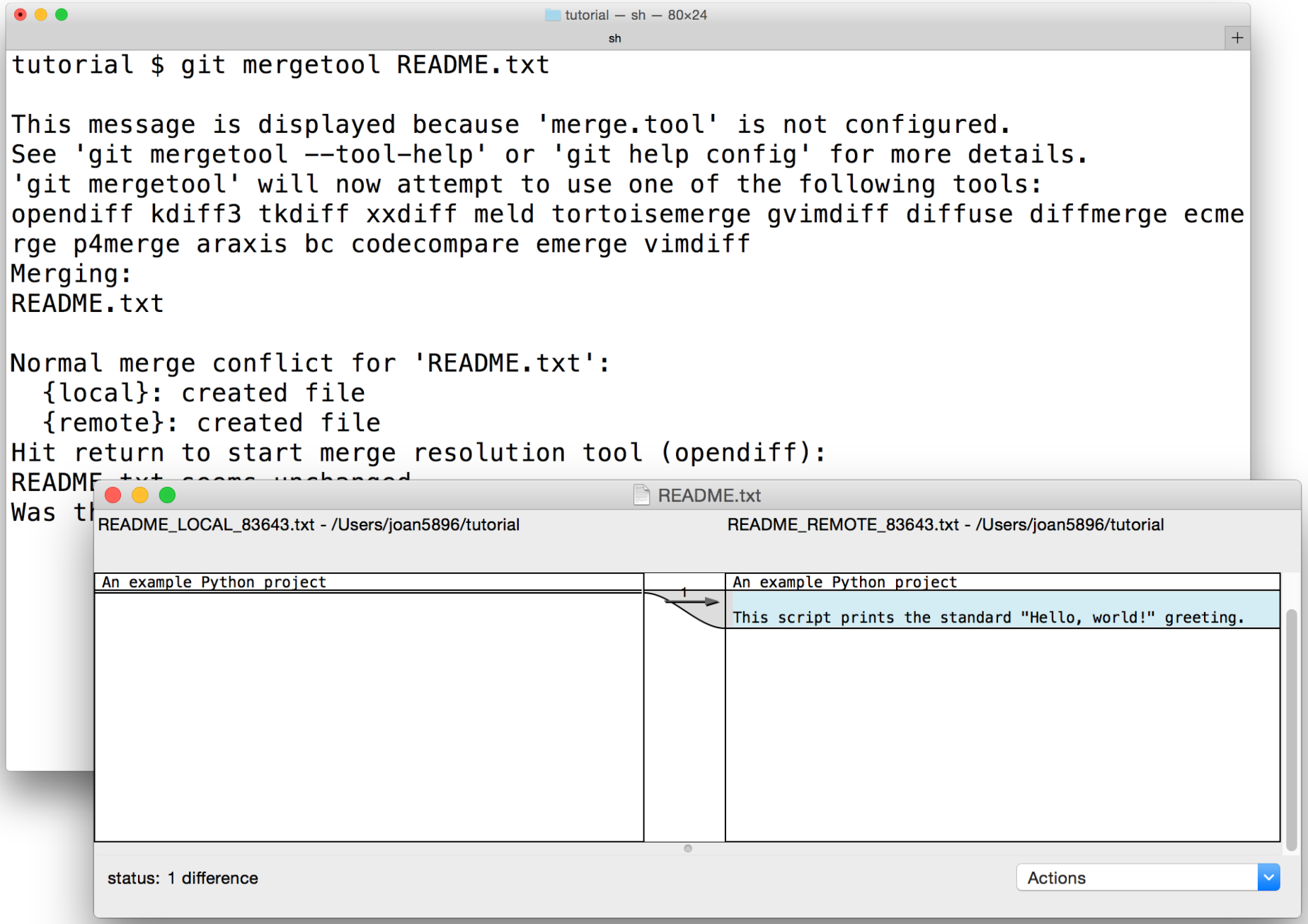
```
$ git fetch <branch> && git merge <branch>
```

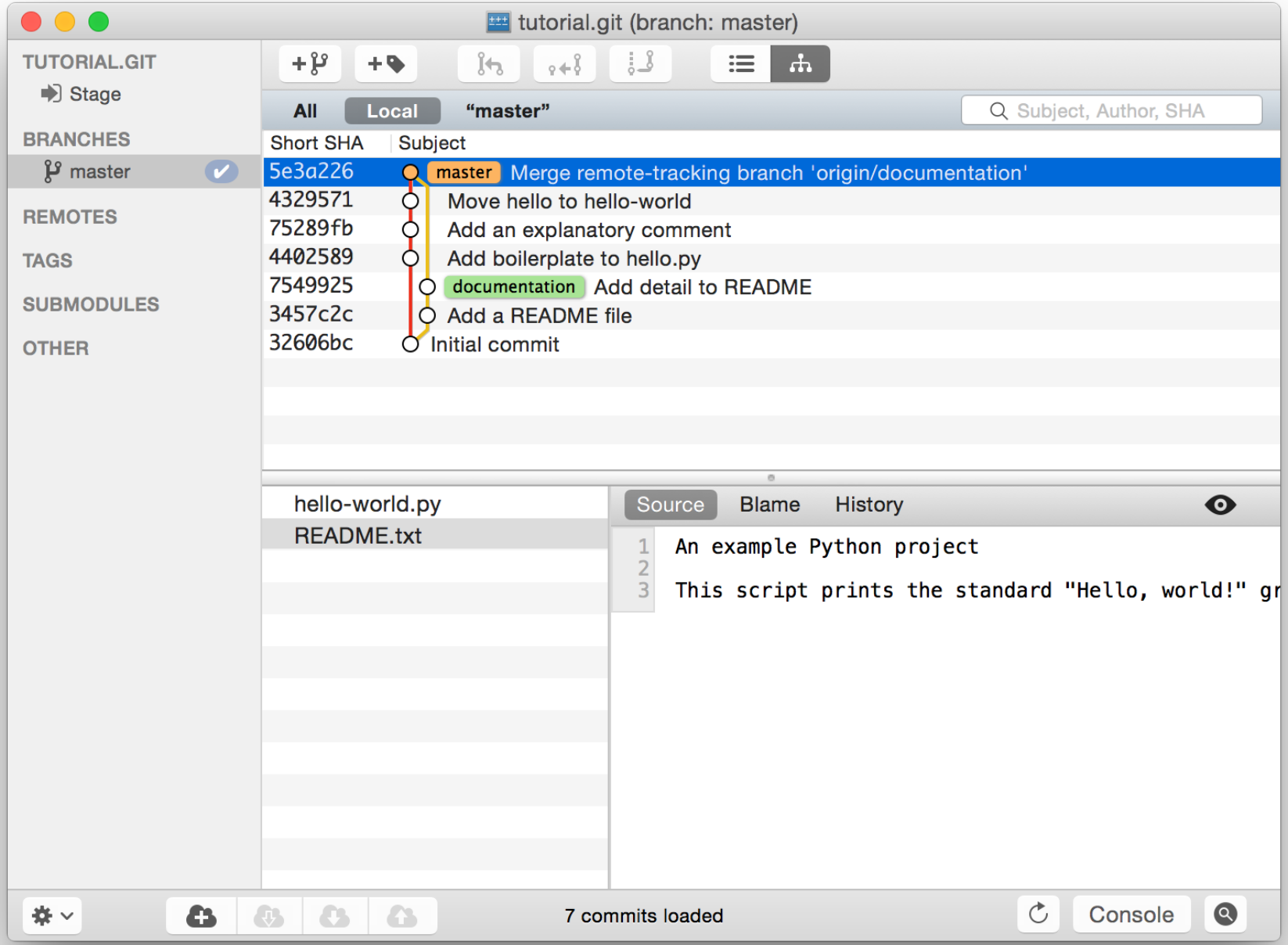

tutorial — bash — 80×24
bash
tutorial $ git reset --hard HEAD^
HEAD is now at 4329571 Move hello to hello-world
tutorial $ echo 'An example Python project' >README.txt
tutorial $ git add README.txt
tutorial $ git commit --message "Add a README file"
[master 0e90a07] Add a README file
 1 file changed, 1 insertion(+)
 create mode 100644 README.txt
tutorial $


tutorial — bash — 80×24
bash
tutorial $ git reset --hard HEAD^
HEAD is now at 4329571 Move hello to hello-world
tutorial $ echo 'An example Python project' >README.txt
tutorial $ git add README.txt
tutorial $ git commit --message "Add a README file"
[master 0e90a07] Add a README file
 1 file changed, 1 insertion(+)
 create mode 100644 README.txt
tutorial $ git merge origin/documentation
Auto-merging README.txt
CONFLICT (add/add): Merge conflict in README.txt
Automatic merge failed; fix conflicts and then commit the result.
tutorial $ cat README.txt
An example Python project
<<<<<<< HEAD
=======

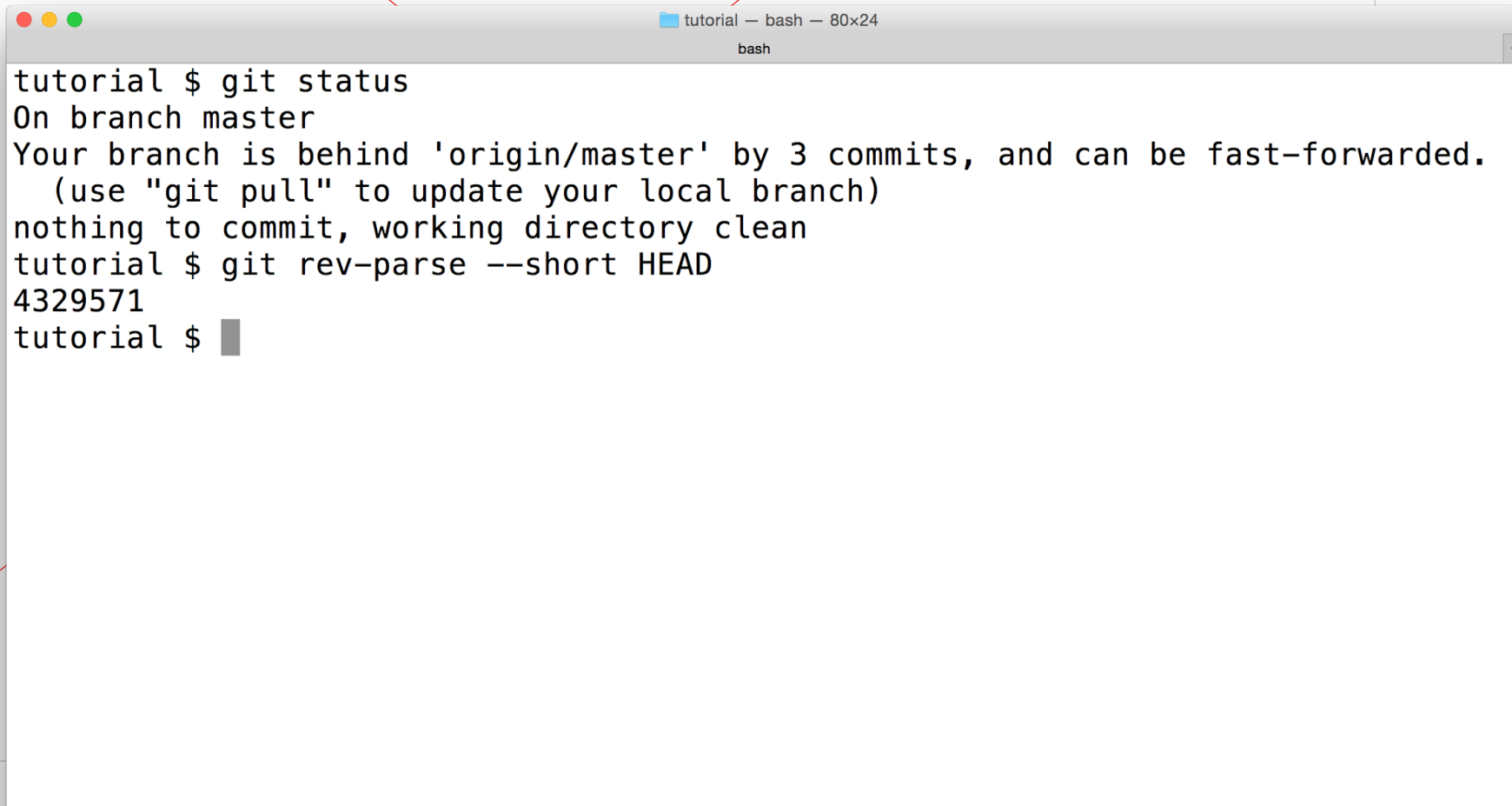
This script prints the standard "Hello, world!" greeting.
>>>>>>> origin/documentation
tutorial $

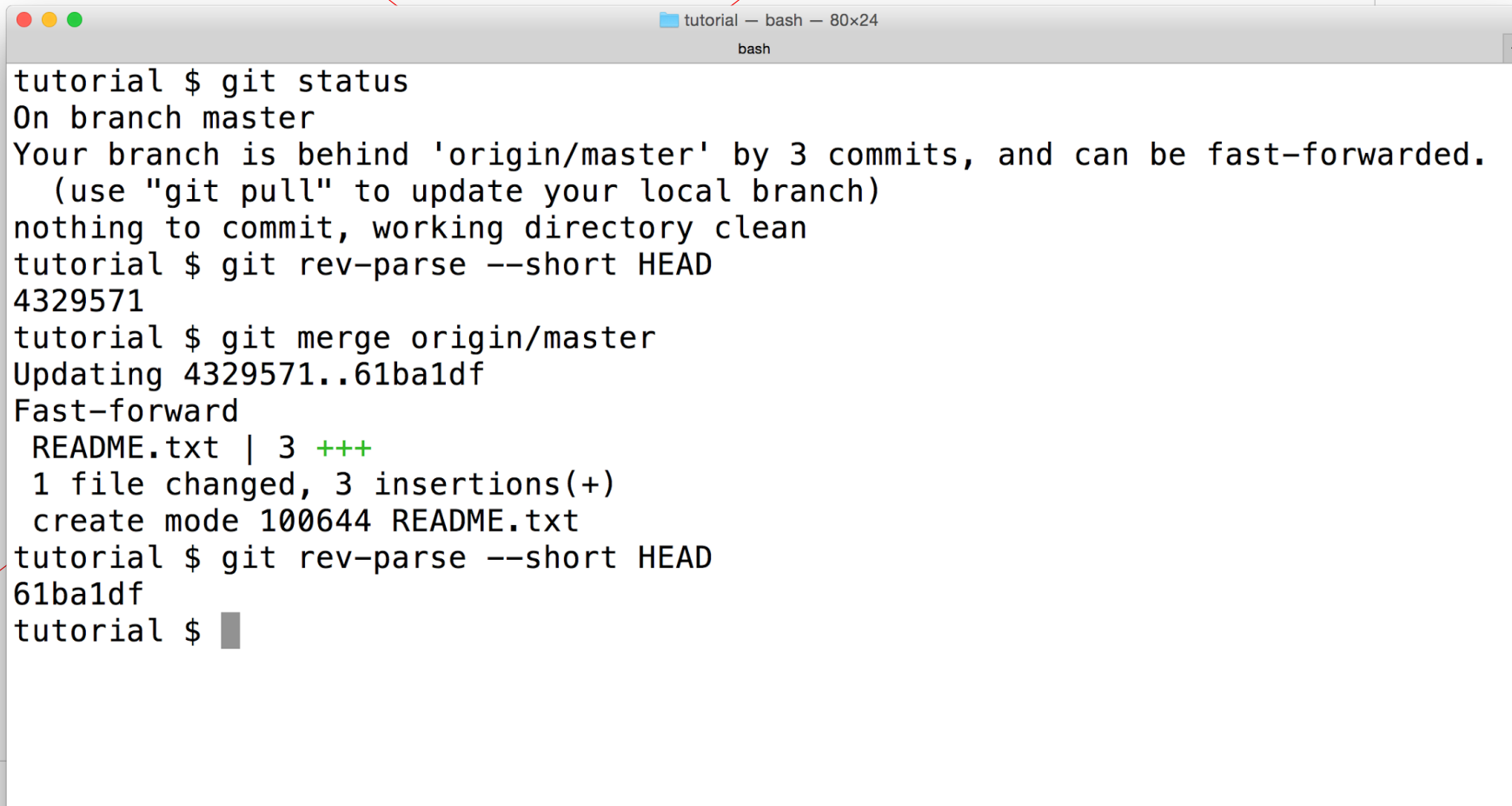

tutorial — bash — 80×24
bash
tutorial $ vi README.txt
tutorial $ cat README.txt
An example Python project
This script prints the standard "Hello, world!" greeting.
tutorial $ git commit --all
[master 775eccb] Merge remote-tracking branch 'origin/documentation'
tutorial $


tutorial — sh — 80×24
sh
tutorial $ git mergetool README.txt
This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecme
rge p4merge araxis bc codecompare emerge vimdiff
Merging:
README.txt
Normal merge conflict for 'README.txt':
{local}: created file
{remote}: created file
Hit return to start merge resolution tool (opendiff):
README.txt seems unchanged.
Was th
README.txt
README_LOCAL_83643.txt - /Users/joan5896/tutorial
README_REMOTE_83643.txt - /Users/joan5896/tutorial
An example Python project
1
An example Python project
This script prints the standard "Hello, world!" greeting.
status: 1 difference
Actions

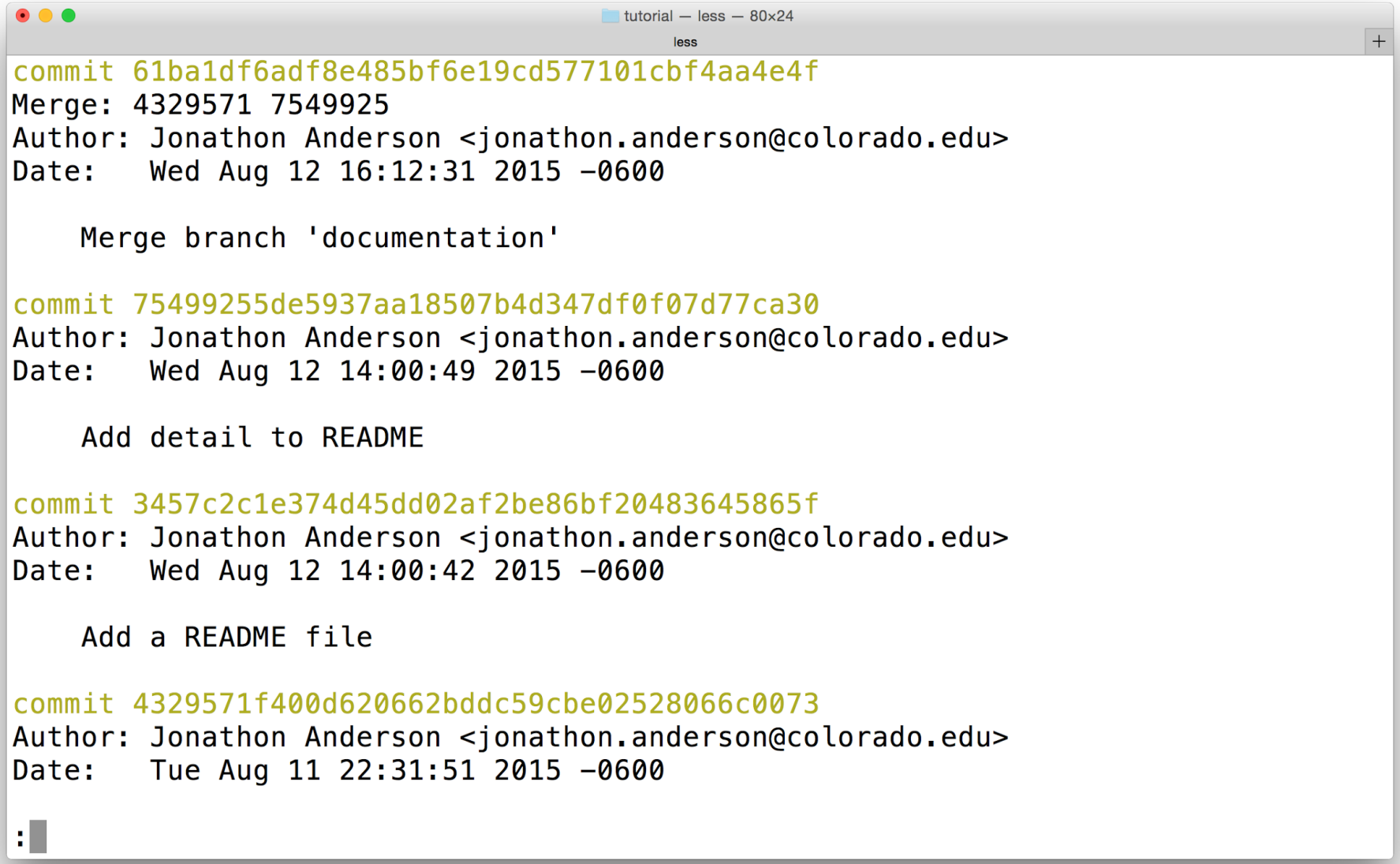

tutorial.git (branch: master)
TUTORIAL.GIT
Stage
BRANCHES
master
REMOTES
TAGS
SUBMODULES
OTHER
All
Local
“master”
Subject, Author, SHA
Short SHA
Subject
5e3a226
master
Merge remote-tracking branch 'origin/documentation'
4329571
Move hello to hello-world
75289fb
Add an explanatory comment
4402589
Add boilerplate to hello.py
7549925
documentation
Add detail to README
3457c2c
Add a README file
32606bc
Initial commit
hello-world.py
README.txt
Source
Blame
History
1 An example Python project
2
3 This script prints the standard "Hello, world!" gr
7 commits loaded
Console

Merge commit:
A commit with multiple parents

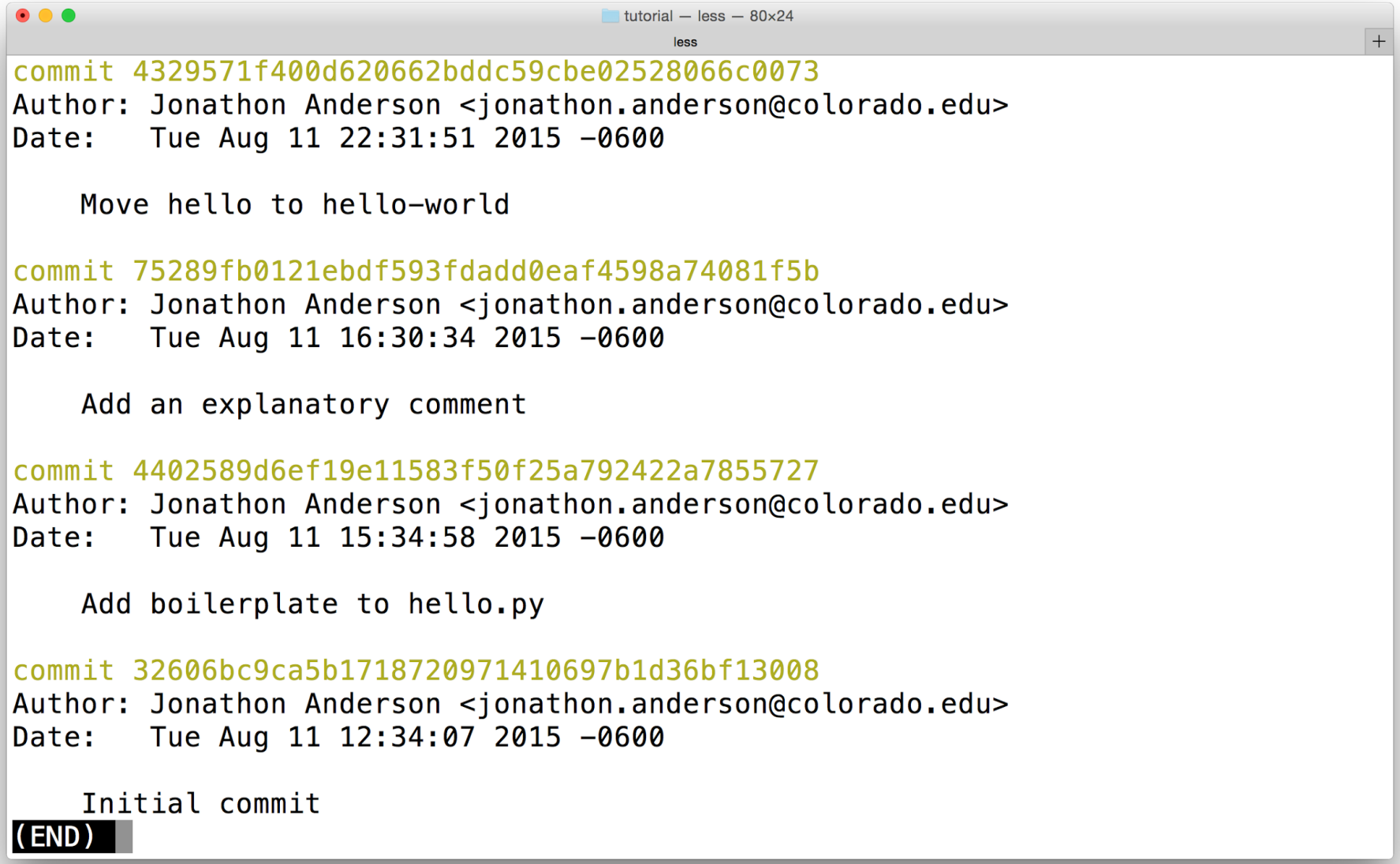

tutorial — bash — 80×24
bash
tutorial $ git status
On branch master
Your branch is behind 'origin/master' by 3 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
nothing to commit, working directory clean
tutorial $ git rev-parse --short HEAD
4329571
tutorial $

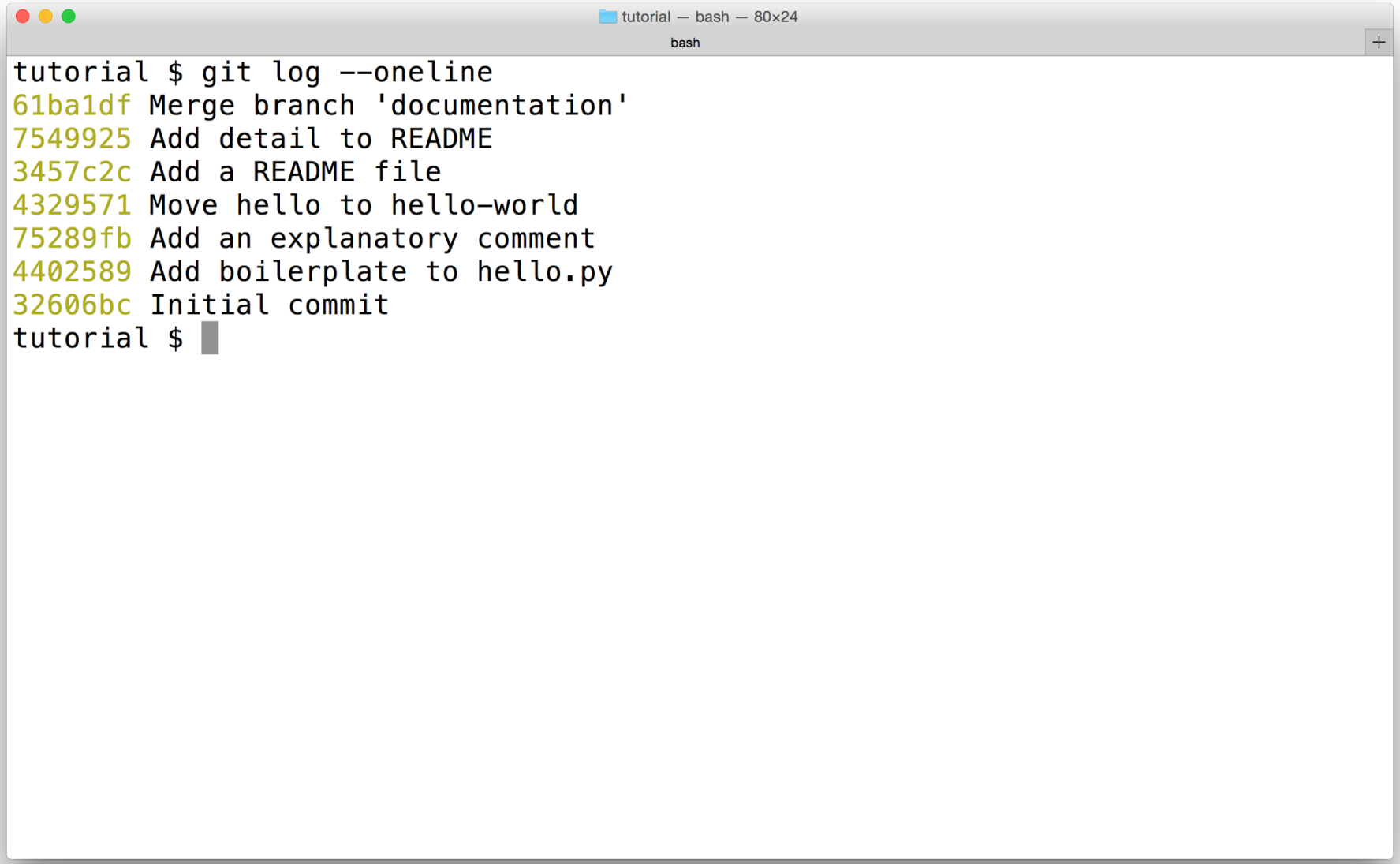

tutorial — bash — 80×24
bash
tutorial $ git status
On branch master
Your branch is behind 'origin/master' by 3 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
nothing to commit, working directory clean
tutorial $ git rev-parse --short HEAD
4329571
tutorial $ git merge origin/master
Updating 4329571..61ba1df
Fast-forward
 README.txt | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 README.txt
tutorial $ git rev-parse --short HEAD
61ba1df
tutorial $

Viewing the history

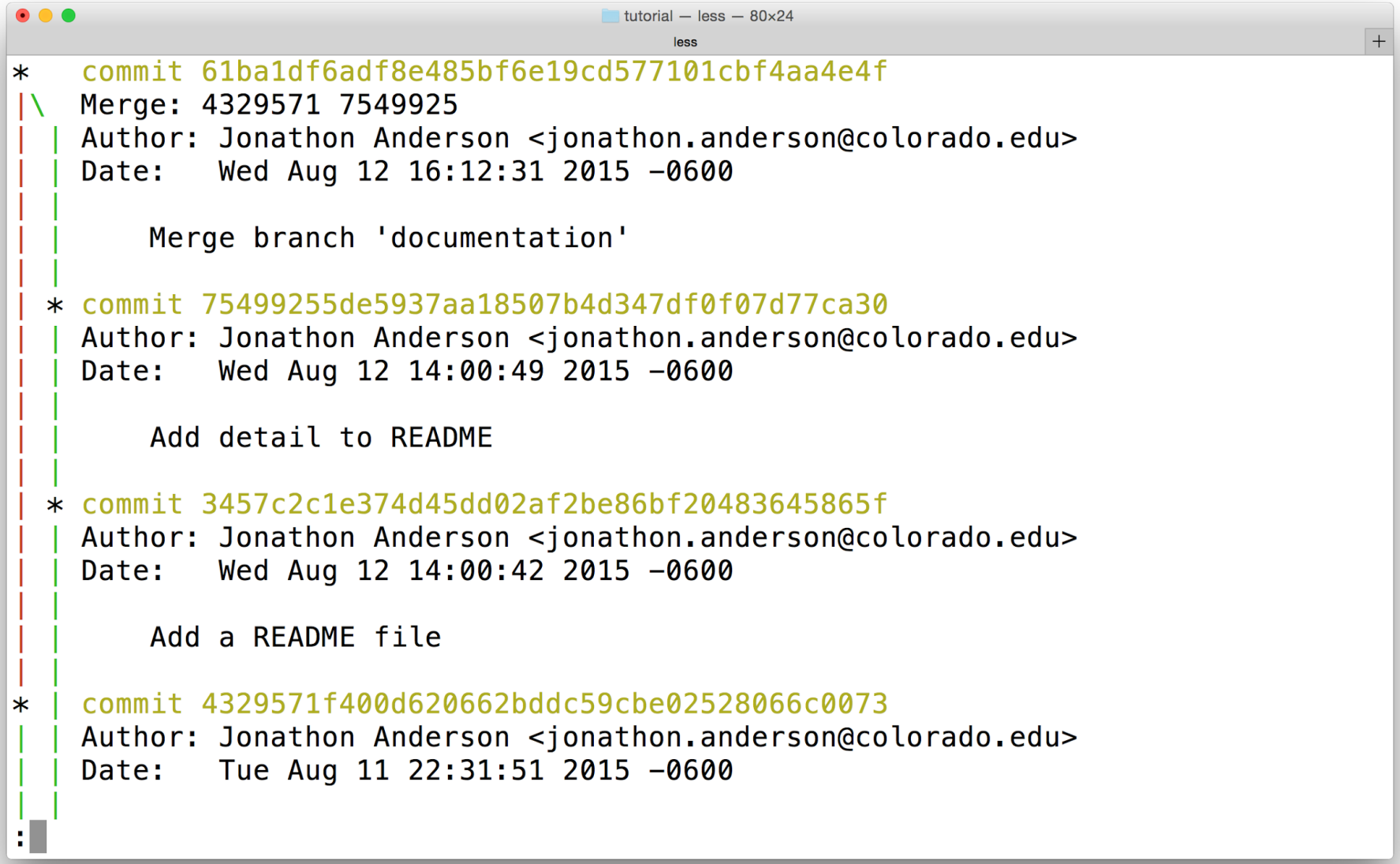

tutorial — less — 80×24
less
commit 61ba1df6adf8e485bf6e19cd577101cbf4aa4e4f
Merge: 4329571 7549925
Author: Jonathon Anderson <jonathon.anderson@colorado.edu>
Date: Wed Aug 12 16:12:31 2015 -0600
Merge branch 'documentation'
commit 75499255de5937aa18507b4d347df0f07d77ca30
Author: Jonathon Anderson <jonathon.anderson@colorado.edu>
Date: Wed Aug 12 14:00:49 2015 -0600
Add detail to README
commit 3457c2c1e374d45dd02af2be86bf20483645865f
Author: Jonathon Anderson <jonathon.anderson@colorado.edu>
Date: Wed Aug 12 14:00:42 2015 -0600
Add a README file
commit 4329571f400d620662bddc59cbe02528066c0073
Author: Jonathon Anderson <jonathon.anderson@colorado.edu>
Date: Tue Aug 11 22:31:51 2015 -0600
:

$ git log

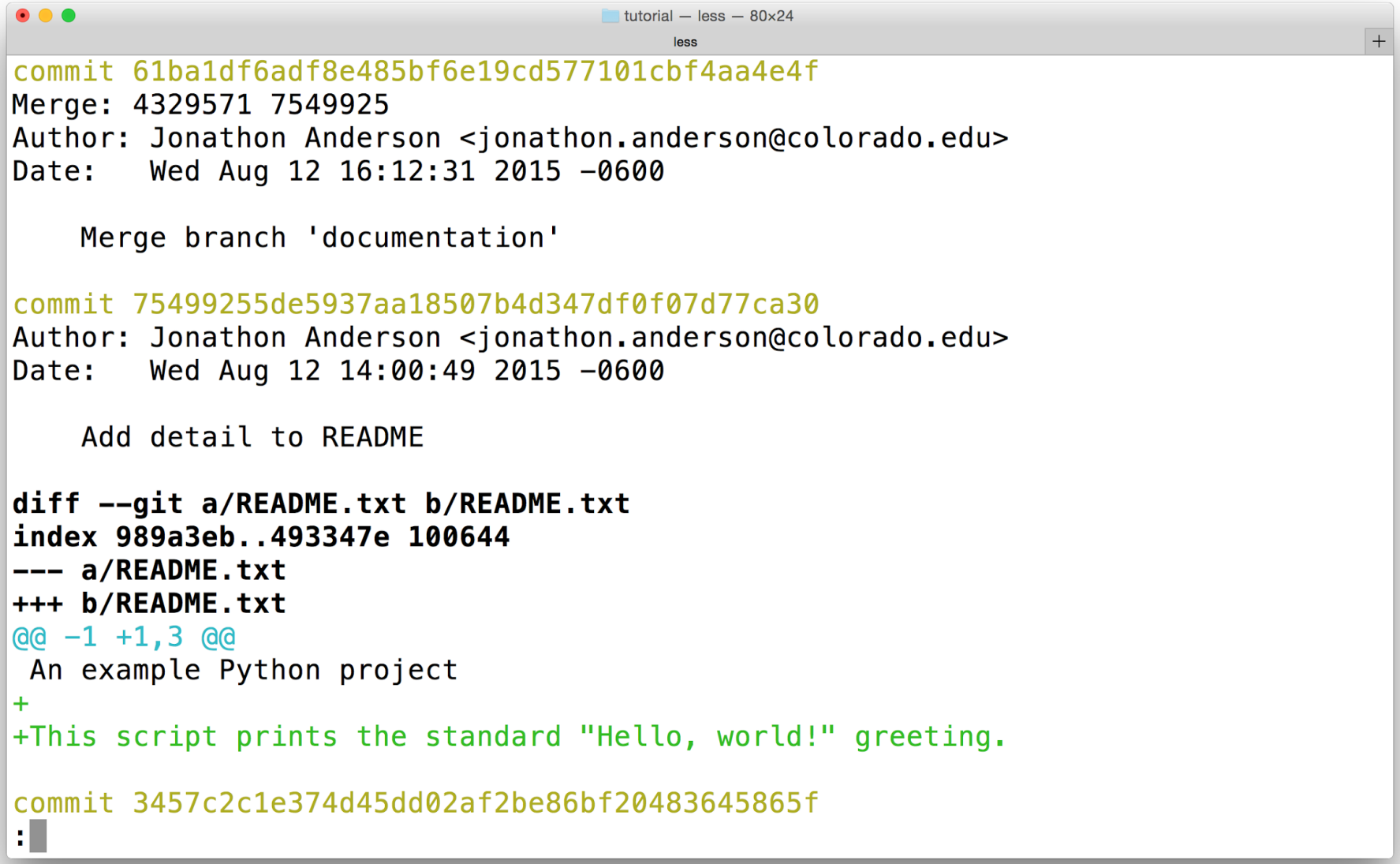

tutorial — less — 80×24
less
commit 4329571f400d620662bddc59cbe02528066c0073
Author: Jonathon Anderson <jonathon.anderson@colorado.edu>
Date: Tue Aug 11 22:31:51 2015 -0600
Move hello to hello-world
commit 75289fb0121ebdf593fdadd0eaf4598a74081f5b
Author: Jonathon Anderson <jonathon.anderson@colorado.edu>
Date: Tue Aug 11 16:30:34 2015 -0600
Add an explanatory comment
commit 4402589d6ef19e11583f50f25a792422a7855727
Author: Jonathon Anderson <jonathon.anderson@colorado.edu>
Date: Tue Aug 11 15:34:58 2015 -0600
Add boilerplate to hello.py
commit 32606bc9ca5b1718720971410697b1d36bf13008
Author: Jonathon Anderson <jonathon.anderson@colorado.edu>
Date: Tue Aug 11 12:34:07 2015 -0600
Initial commit
(END)



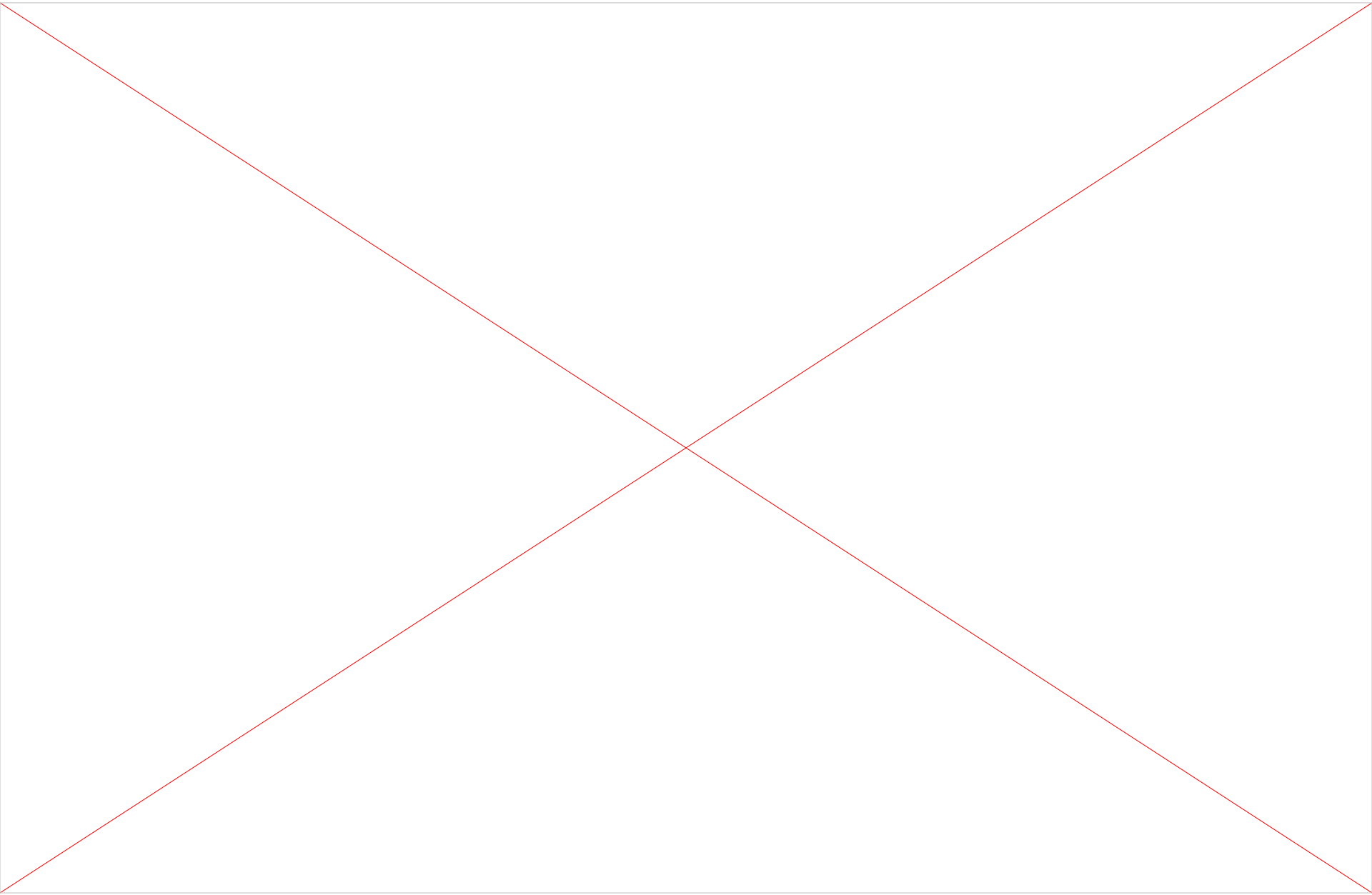
$ git log --oneline



$ git log --graph



$ git log --patch



AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

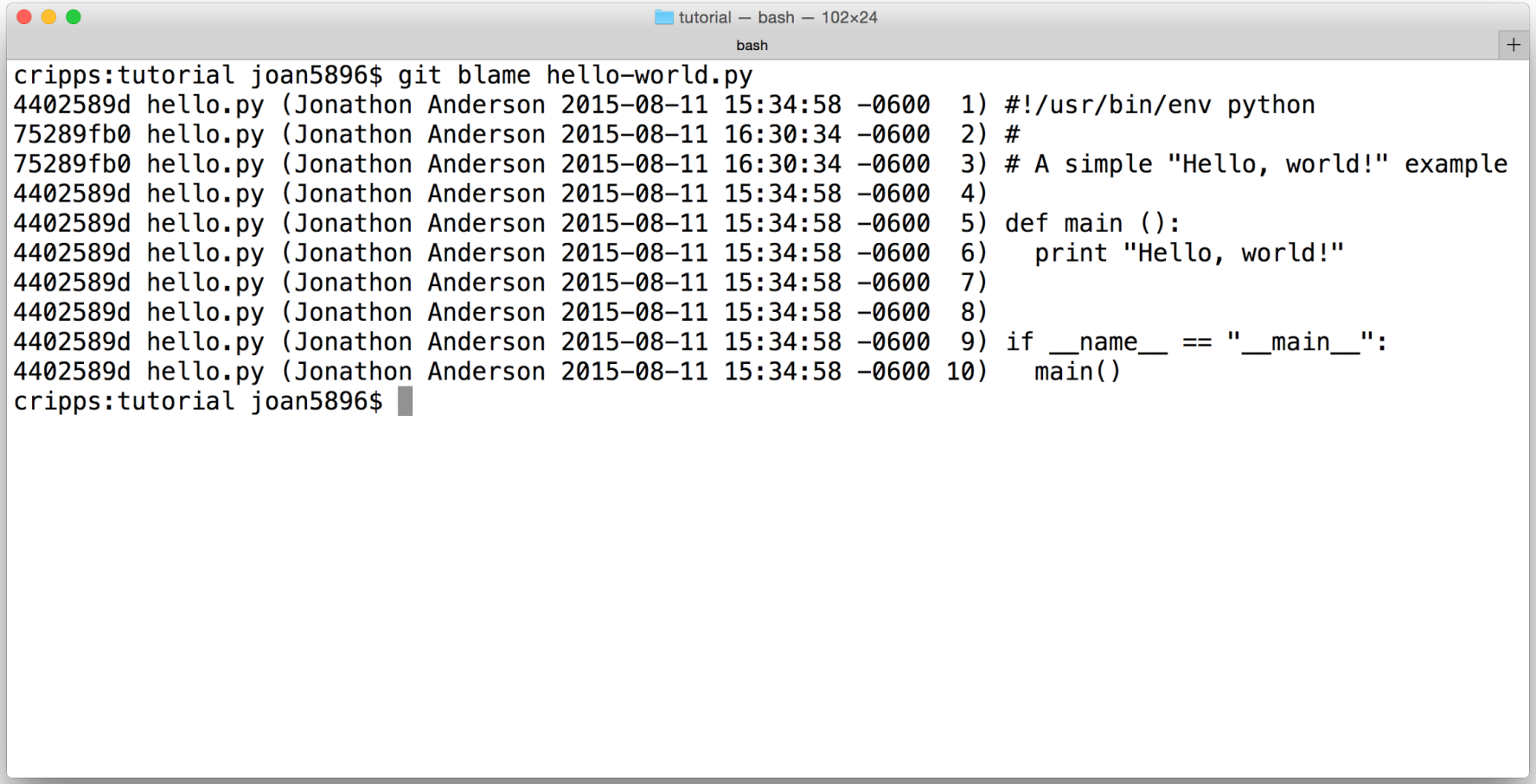
https://xkcd.com/1296/

How to Write a Git Commit Message

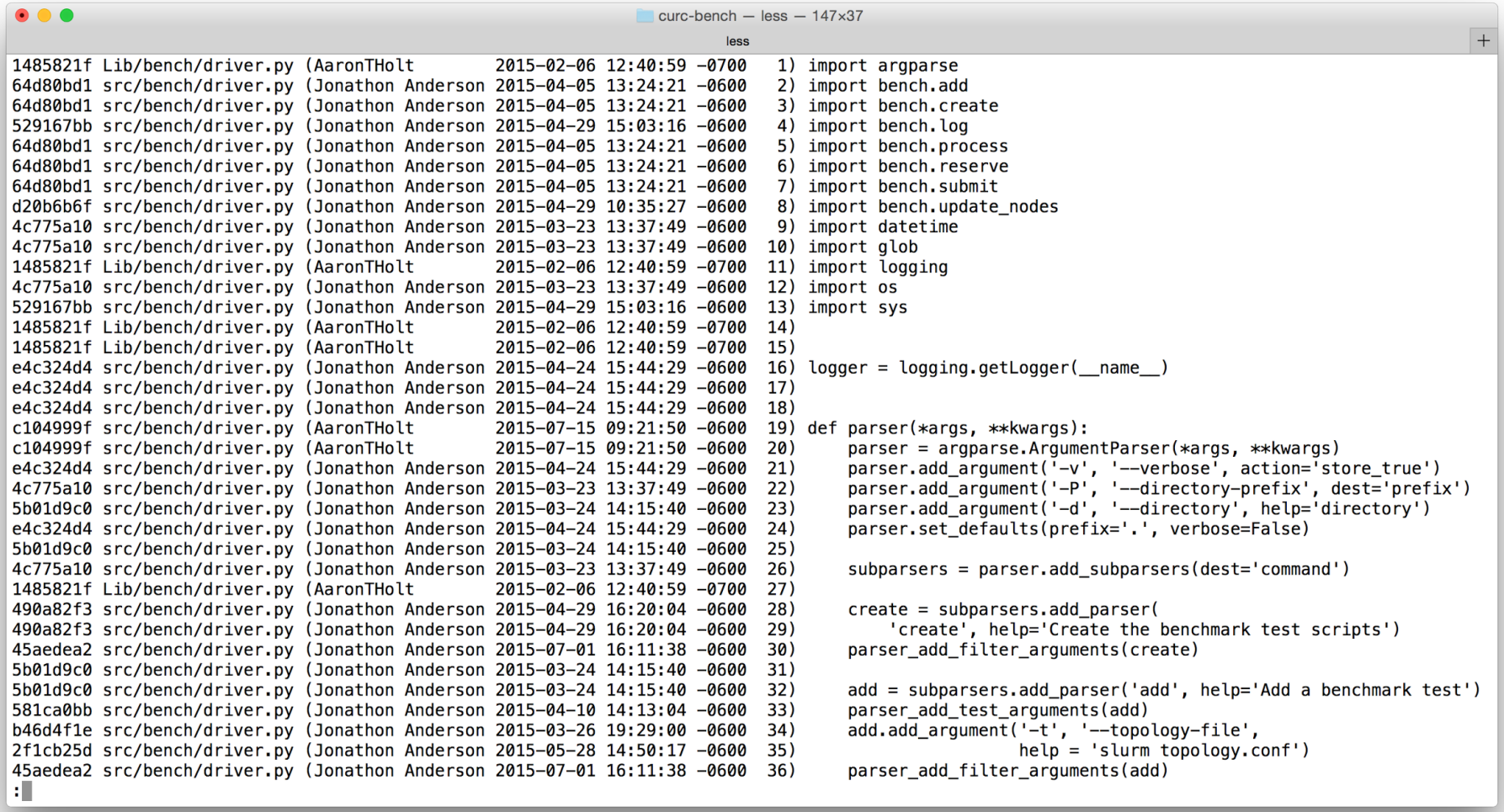
http://chris.beams.io/posts/git-commit/

- Separate subject from body with a blank line
- Limit the subject line to 50 characters
- Capitalize the subject line
- Do not end the subject line with a period
- Use the imperative mood in the subject line
- Wrap the body at 72 characters
- Use the body to explain what and why vs. how

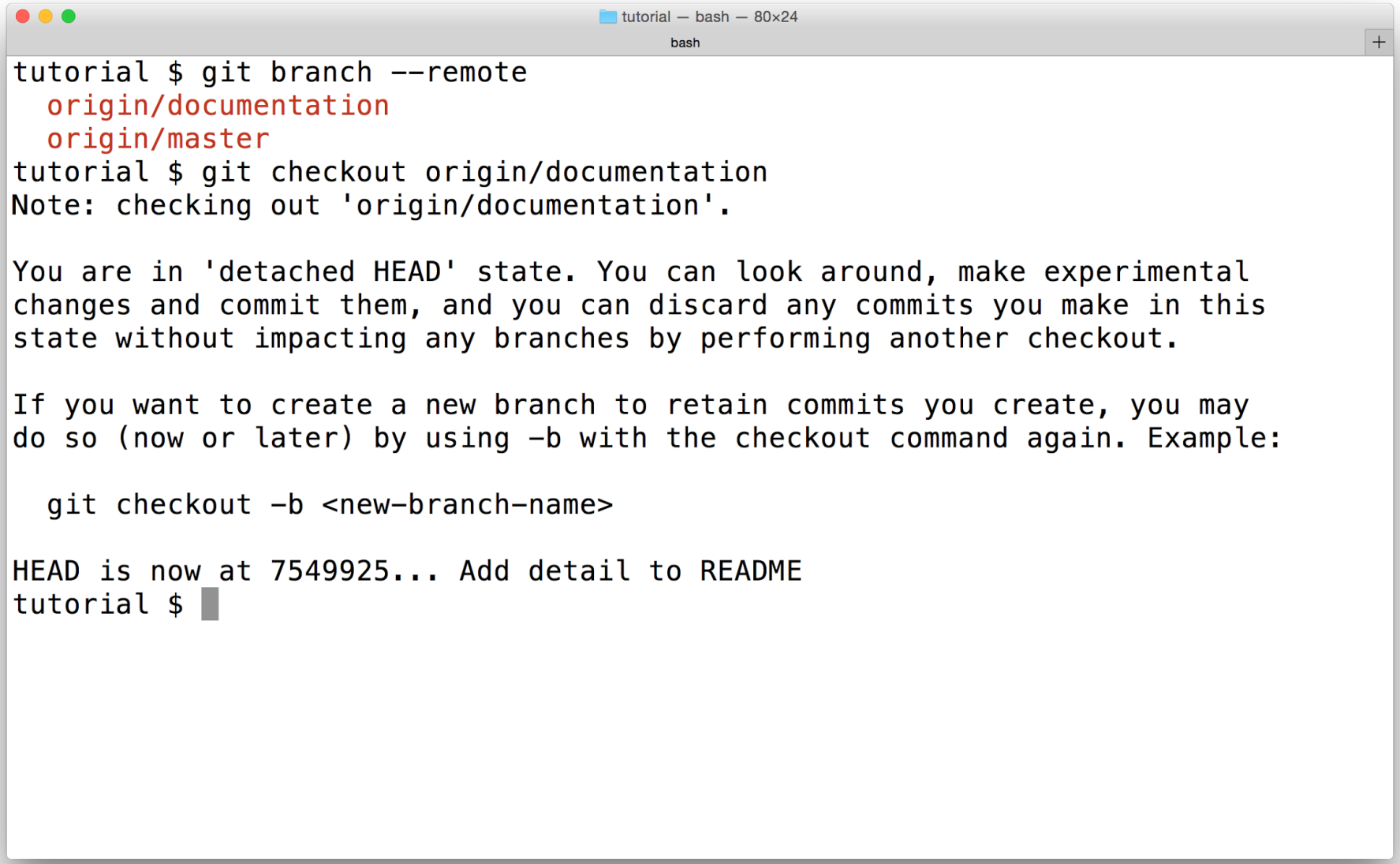
http://chris.beams.io/posts/git-commit/



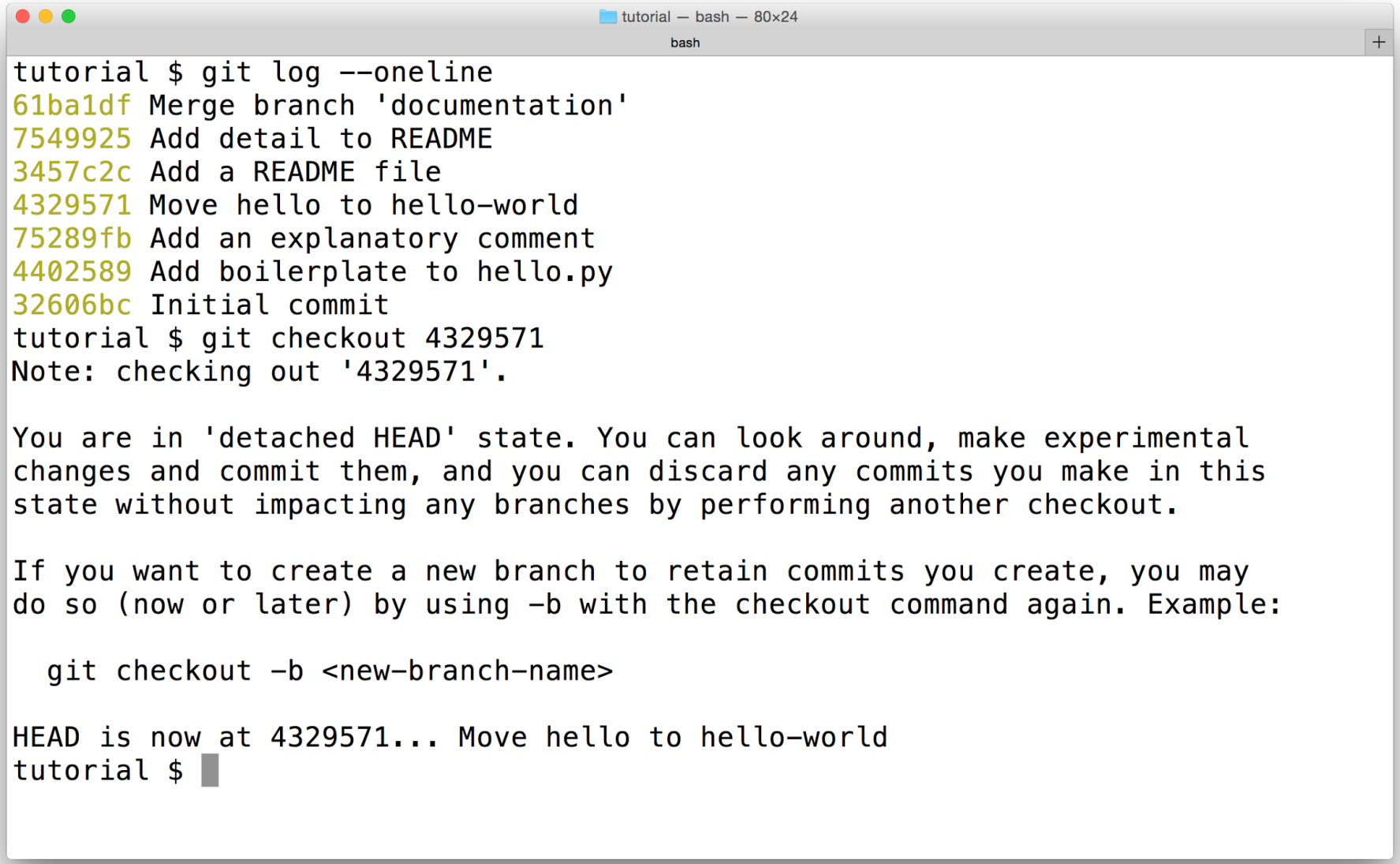
$ git blame <file>


curc-bench — less — 147×37
less
1485821f Lib/bench/driver.py (AaronTHolt 2015-02-06 12:40:59 -0700 1) import argparse
64d80bd1 src/bench/driver.py (Jonathon Anderson 2015-04-05 13:24:21 -0600 2) import bench.add
64d80bd1 src/bench/driver.py (Jonathon Anderson 2015-04-05 13:24:21 -0600 3) import bench.create
529167bb src/bench/driver.py (Jonathon Anderson 2015-04-29 15:03:16 -0600 4) import bench.log
64d80bd1 src/bench/driver.py (Jonathon Anderson 2015-04-05 13:24:21 -0600 5) import bench.process
64d80bd1 src/bench/driver.py (Jonathon Anderson 2015-04-05 13:24:21 -0600 6) import bench.reserve
64d80bd1 src/bench/driver.py (Jonathon Anderson 2015-04-05 13:24:21 -0600 7) import bench.submit
d20b6b6f src/bench/driver.py (Jonathon Anderson 2015-04-29 10:35:27 -0600 8) import bench.update_nodes
4c775a10 src/bench/driver.py (Jonathon Anderson 2015-03-23 13:37:49 -0600 9) import datetime
4c775a10 src/bench/driver.py (Jonathon Anderson 2015-03-23 13:37:49 -0600 10) import glob
1485821f Lib/bench/driver.py (AaronTHolt 2015-02-06 12:40:59 -0700 11) import logging
4c775a10 src/bench/driver.py (Jonathon Anderson 2015-03-23 13:37:49 -0600 12) import os
529167bb src/bench/driver.py (Jonathon Anderson 2015-04-29 15:03:16 -0600 13) import sys
1485821f Lib/bench/driver.py (AaronTHolt 2015-02-06 12:40:59 -0700 14)
1485821f Lib/bench/driver.py (AaronTHolt 2015-02-06 12:40:59 -0700 15)
e4c324d4 src/bench/driver.py (Jonathon Anderson 2015-04-24 15:44:29 -0600 16) logger = logging.getLogger(__name__)
e4c324d4 src/bench/driver.py (Jonathon Anderson 2015-04-24 15:44:29 -0600 17)
e4c324d4 src/bench/driver.py (Jonathon Anderson 2015-04-24 15:44:29 -0600 18)
c104999f src/bench/driver.py (AaronTHolt 2015-07-15 09:21:50 -0600 19) def parser(*args, **kwargs):
c104999f src/bench/driver.py (AaronTHolt 2015-07-15 09:21:50 -0600 20) parser = argparse.ArgumentParser(*args, **kwargs)
e4c324d4 src/bench/driver.py (Jonathon Anderson 2015-04-24 15:44:29 -0600 21) parser.add_argument('-v', '--verbose', action='store_true')
4c775a10 src/bench/driver.py (Jonathon Anderson 2015-03-23 13:37:49 -0600 22) parser.add_argument('-P', '--directory-prefix', dest='prefix')
5b01d9c0 src/bench/driver.py (Jonathon Anderson 2015-03-24 14:15:40 -0600 23) parser.add_argument('-d', '--directory', help='directory')
e4c324d4 src/bench/driver.py (Jonathon Anderson 2015-04-24 15:44:29 -0600 24) parser.set_defaults(prefix='.', verbose=False)
5b01d9c0 src/bench/driver.py (Jonathon Anderson 2015-03-24 14:15:40 -0600 25)
4c775a10 src/bench/driver.py (Jonathon Anderson 2015-03-23 13:37:49 -0600 26) subparsers = parser.add_subparsers(dest='command')
1485821f Lib/bench/driver.py (AaronTHolt 2015-02-06 12:40:59 -0700 27)
490a82f3 src/bench/driver.py (Jonathon Anderson 2015-04-29 16:20:04 -0600 28) create = subparsers.add_parser(
490a82f3 src/bench/driver.py (Jonathon Anderson 2015-04-29 16:20:04 -0600 29) 'create', help='Create the benchmark test scripts')
45aedea2 src/bench/driver.py (Jonathon Anderson 2015-07-01 16:11:38 -0600 30) parser_add_filter_arguments(create)
5b01d9c0 src/bench/driver.py (Jonathon Anderson 2015-03-24 14:15:40 -0600 31)
5b01d9c0 src/bench/driver.py (Jonathon Anderson 2015-03-24 14:15:40 -0600 32) add = subparsers.add_parser('add', help='Add a benchmark test')
581ca0bb src/bench/driver.py (Jonathon Anderson 2015-04-10 14:13:04 -0600 33) parser_add_test_arguments(add)
b46d4f1e src/bench/driver.py (Jonathon Anderson 2015-03-26 19:29:00 -0600 34) add.add_argument('-t', '--topology-file',
2f1cb25d src/bench/driver.py (Jonathon Anderson 2015-05-28 14:50:17 -0600 35) help = 'slurm topology.conf')
45aedea2 src/bench/driver.py (Jonathon Anderson 2015-07-01 16:11:38 -0600 36) parser_add_filter_arguments(add)
:

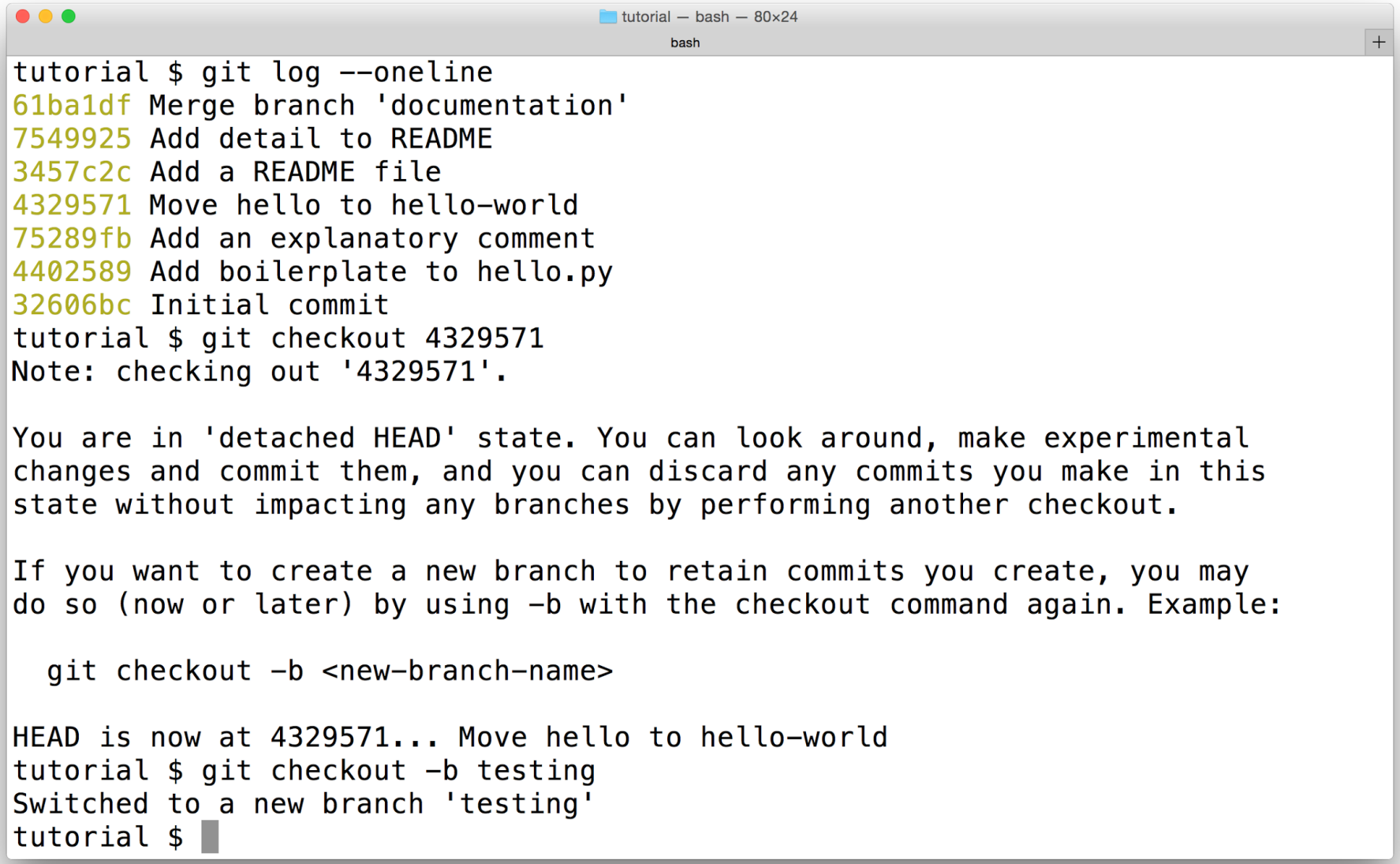
Checkout:
To get content out of the repository



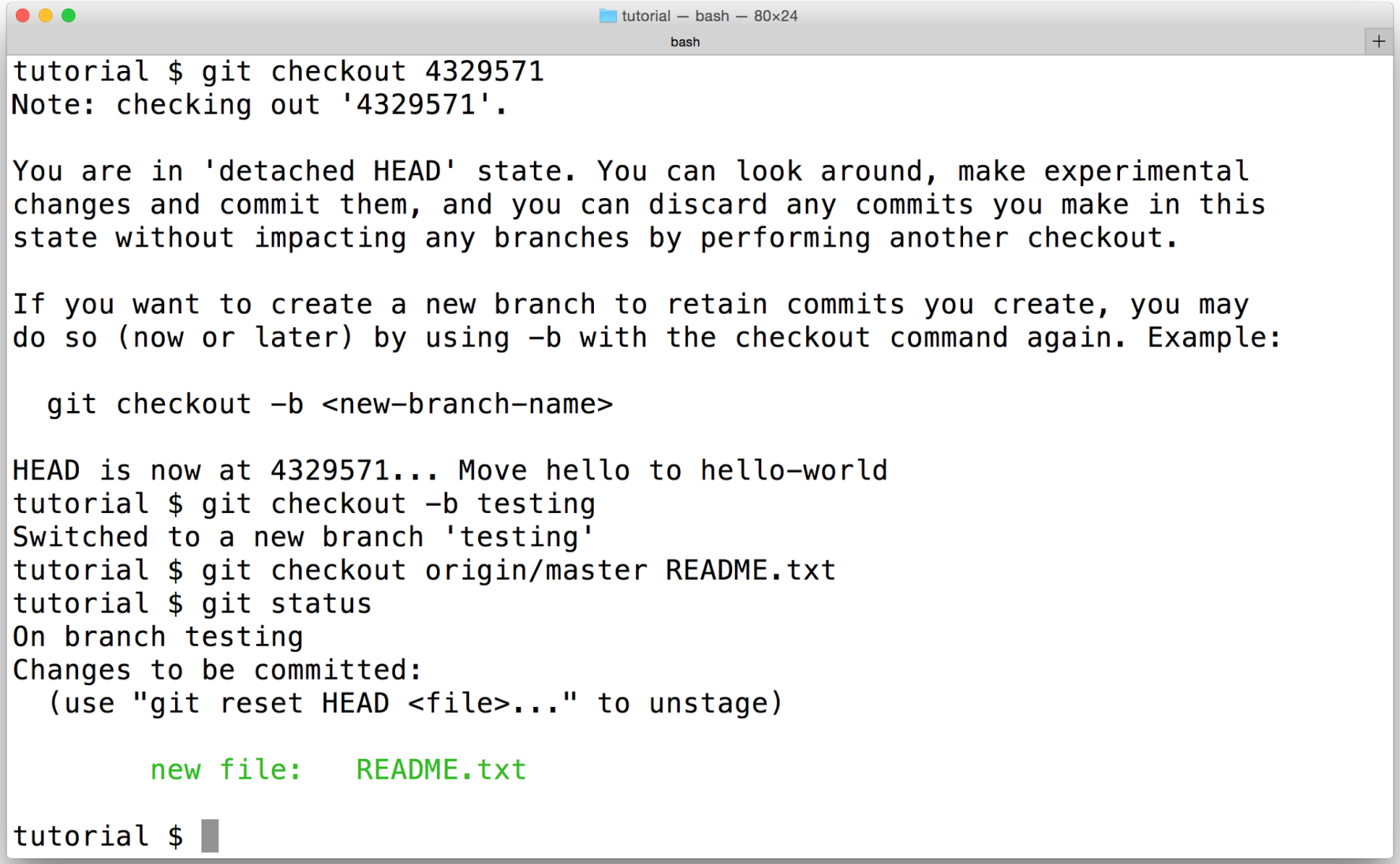
$ git checkout <branch>



$ git checkout <SHA>



$ git checkout -b <new branch>



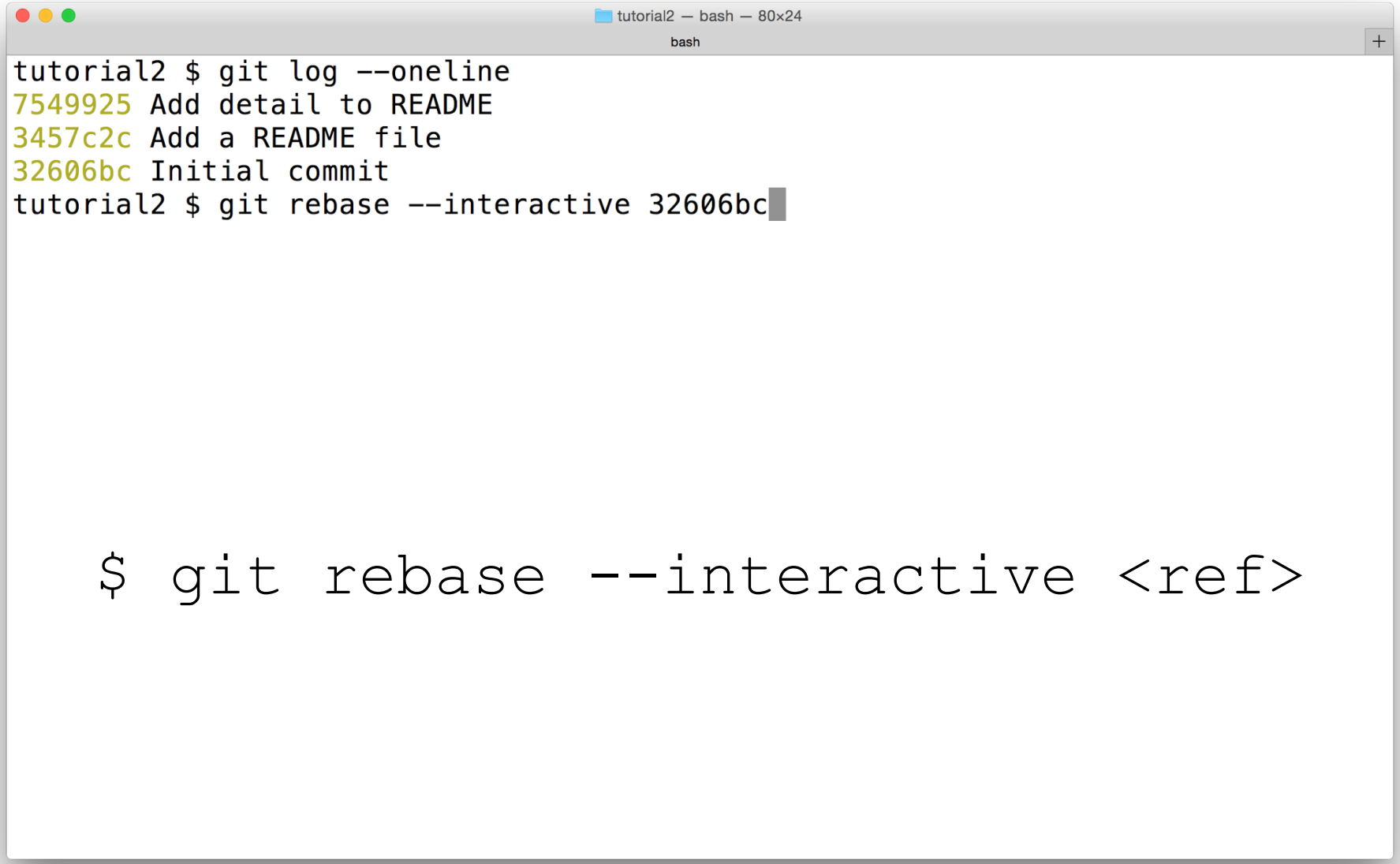
$ git checkout <ref> <file>

Rewriting history;
or:
how to uncommit that private key or bug you just committed

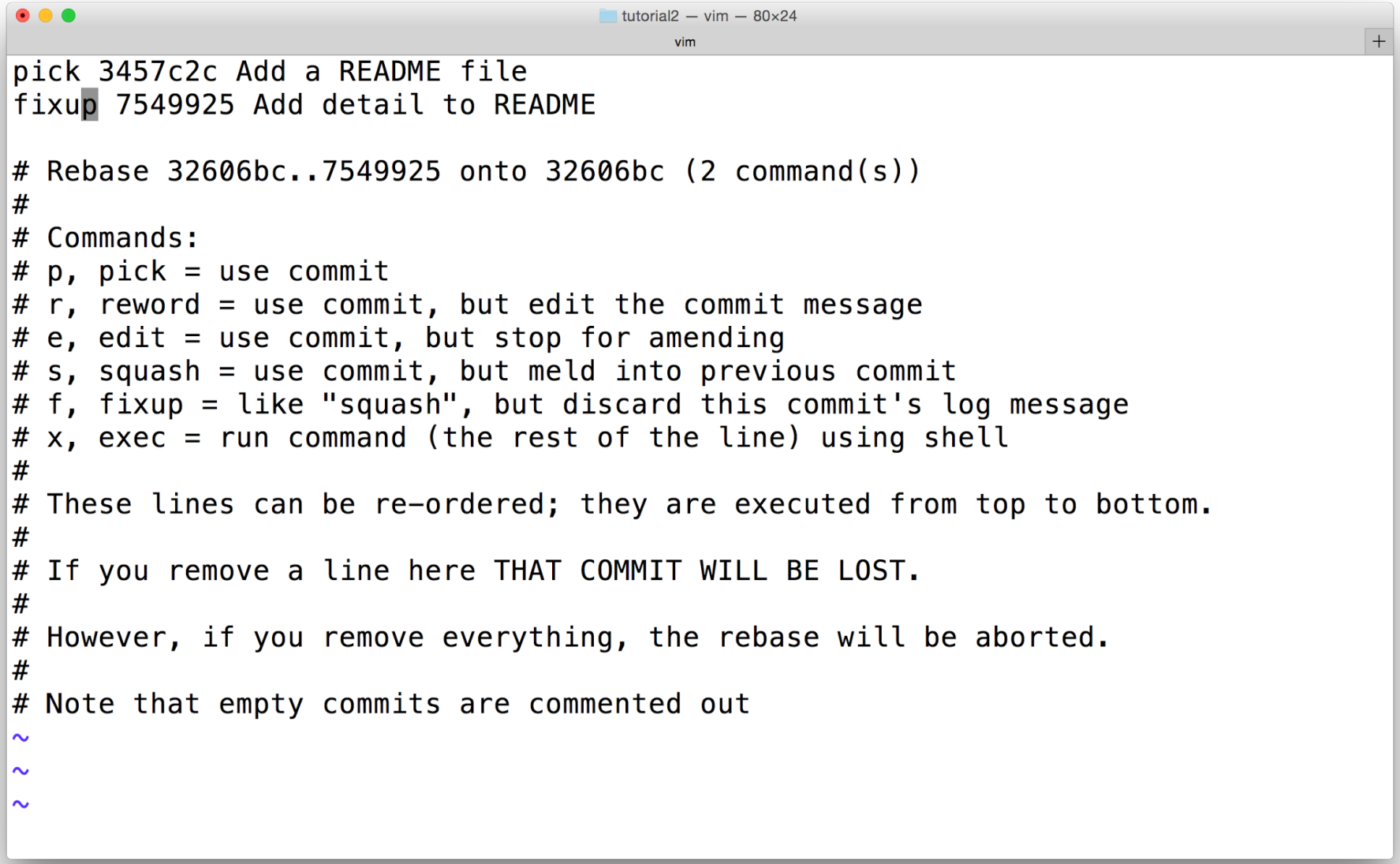
```
$ git rm <file>
```

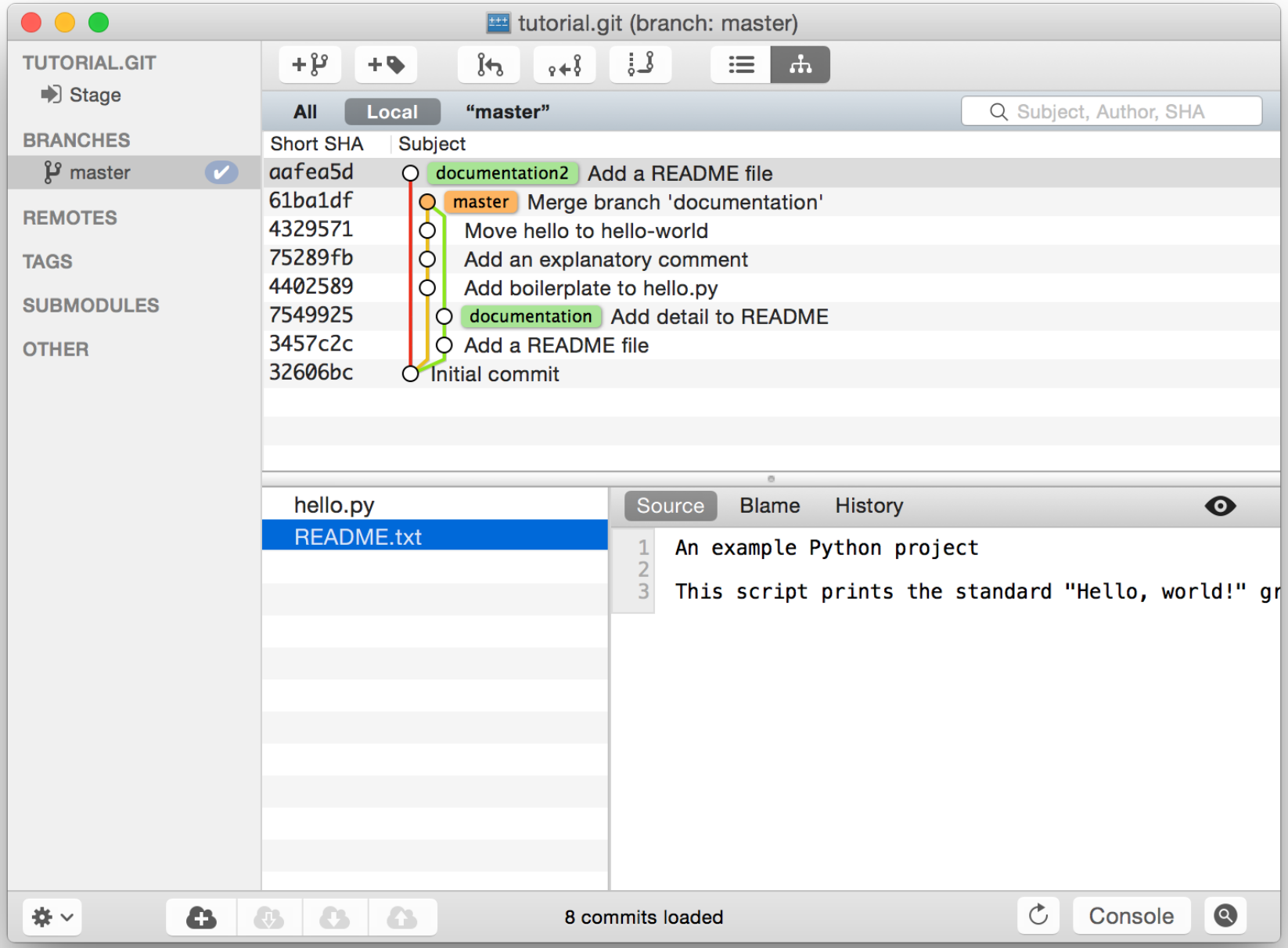
```
$ git revert <commit>
```

```
$ git commit --amend
```



$ git rebase --interactive <ref>


tutorial2 — vim — 80×24
vim
pick 3457c2c Add a README file
fixup 7549925 Add detail to README
Rebase 32606bc..7549925 onto 32606bc (2 command(s))
#
Commands:
p, pick = use commit
r, reword = use commit, but edit the commit message
e, edit = use commit, but stop for amending
s, squash = use commit, but meld into previous commit
f, fixup = like "squash", but discard this commit's log message
x, exec = run command (the rest of the line) using shell
#
These lines can be re-ordered; they are executed from top to bottom.
#
If you remove a line here THAT COMMIT WILL BE LOST.
#
However, if you remove everything, the rebase will be aborted.
#
Note that empty commits are commented out
~
~
~

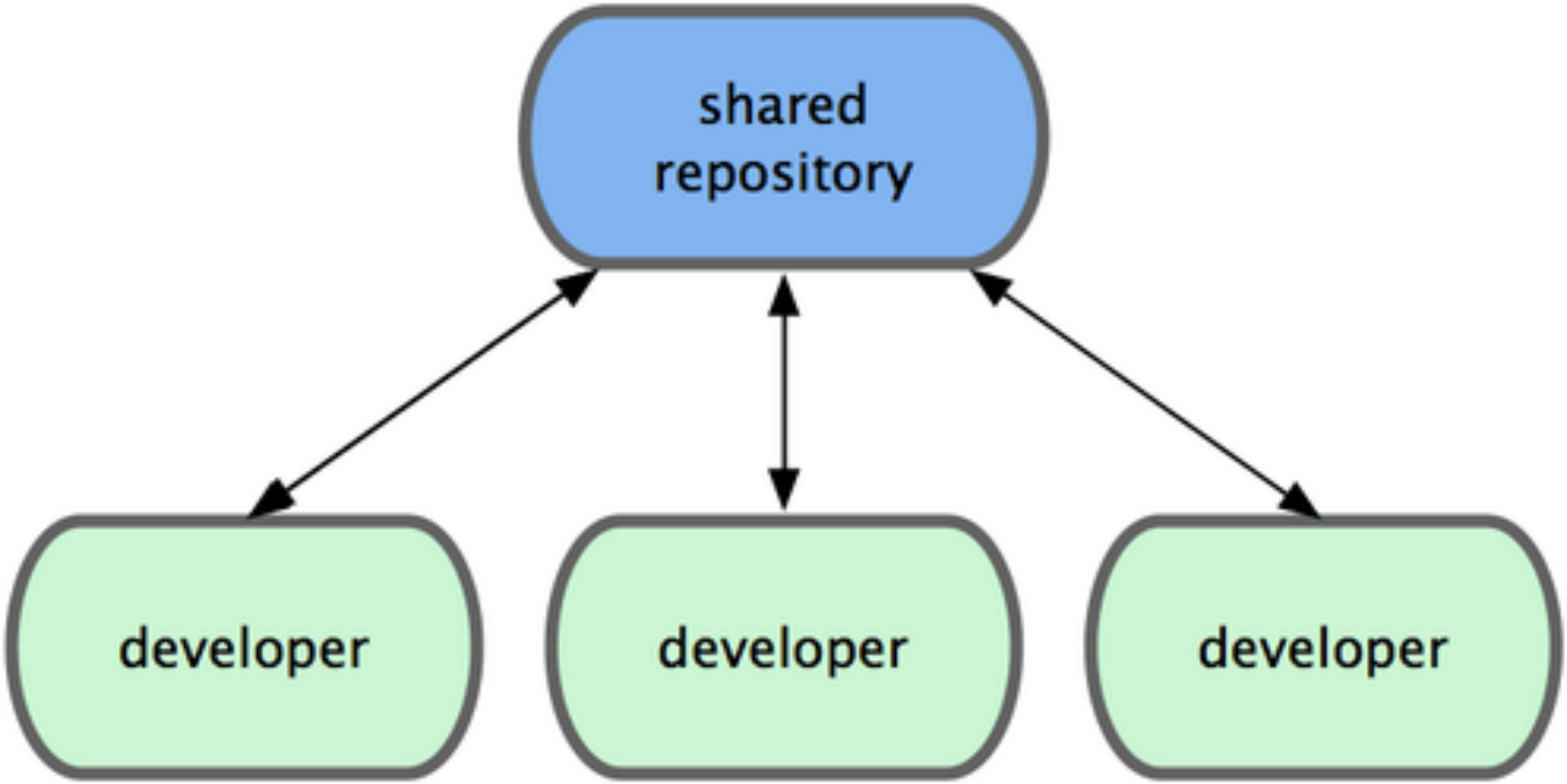

tutorial.git (branch: master)
TUTORIAL.GIT
Stage
BRANCHES
master
REMOTES
TAGS
SUBMODULES
OTHER
All
Local
“master”
Subject, Author, SHA
Short SHA
Subject
aafea5d
documentation2
Add a README file
61ba1df
master
Merge branch 'documentation'
4329571
Move hello to hello-world
75289fb
Add an explanatory comment
4402589
Add boilerplate to hello.py
7549925
documentation
Add detail to README
3457c2c
Add a README file
32606bc
Initial commit
hello.py
README.txt
Source
Blame
History
1 An example Python project
2
3 This script prints the standard "Hello, world!" gr
8 commits loaded
Console

```
$ git push --force <remote> <branch>
```

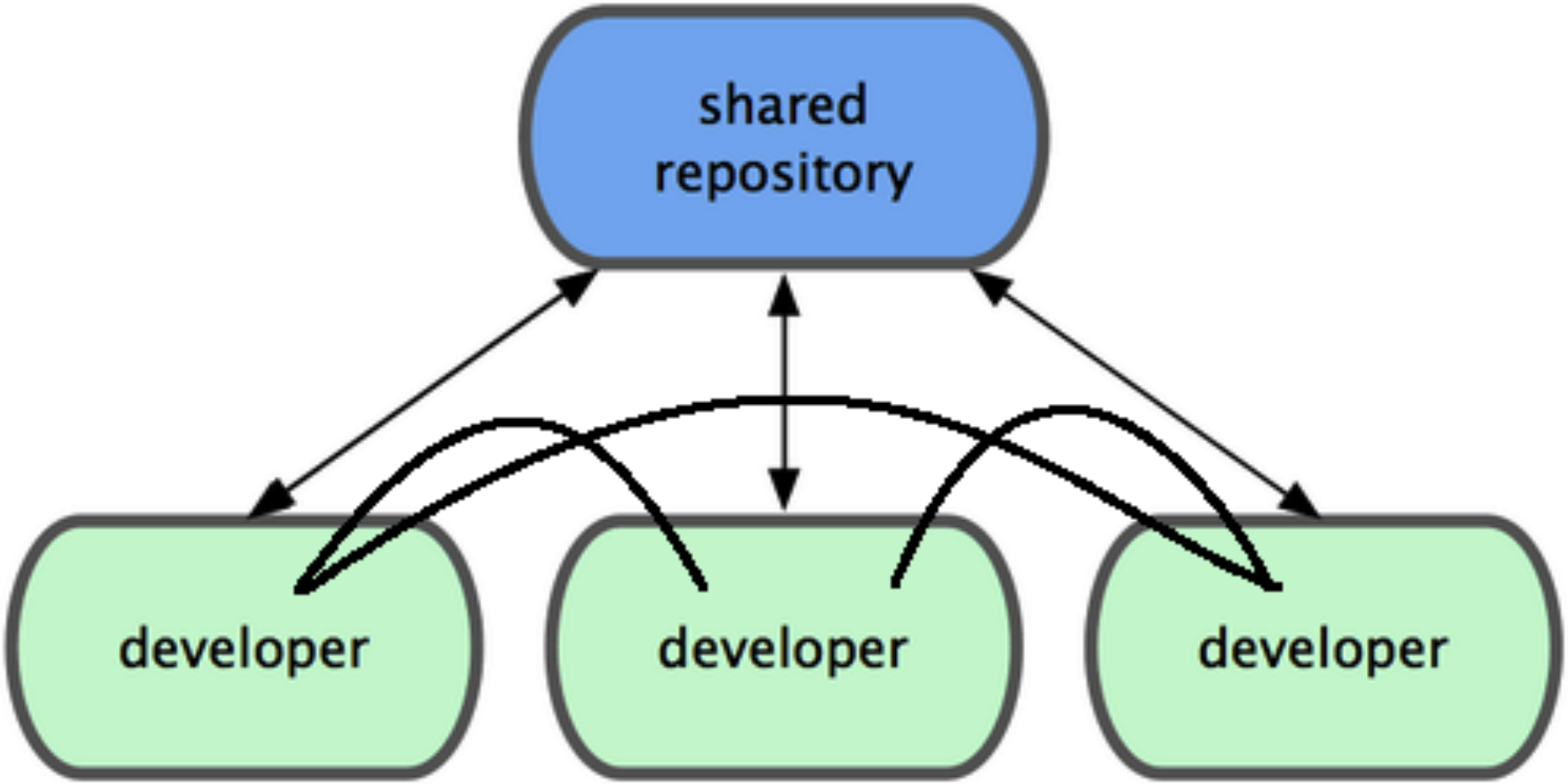
Cleanup your commits *before* pushing

```
$ git filter-branch
```

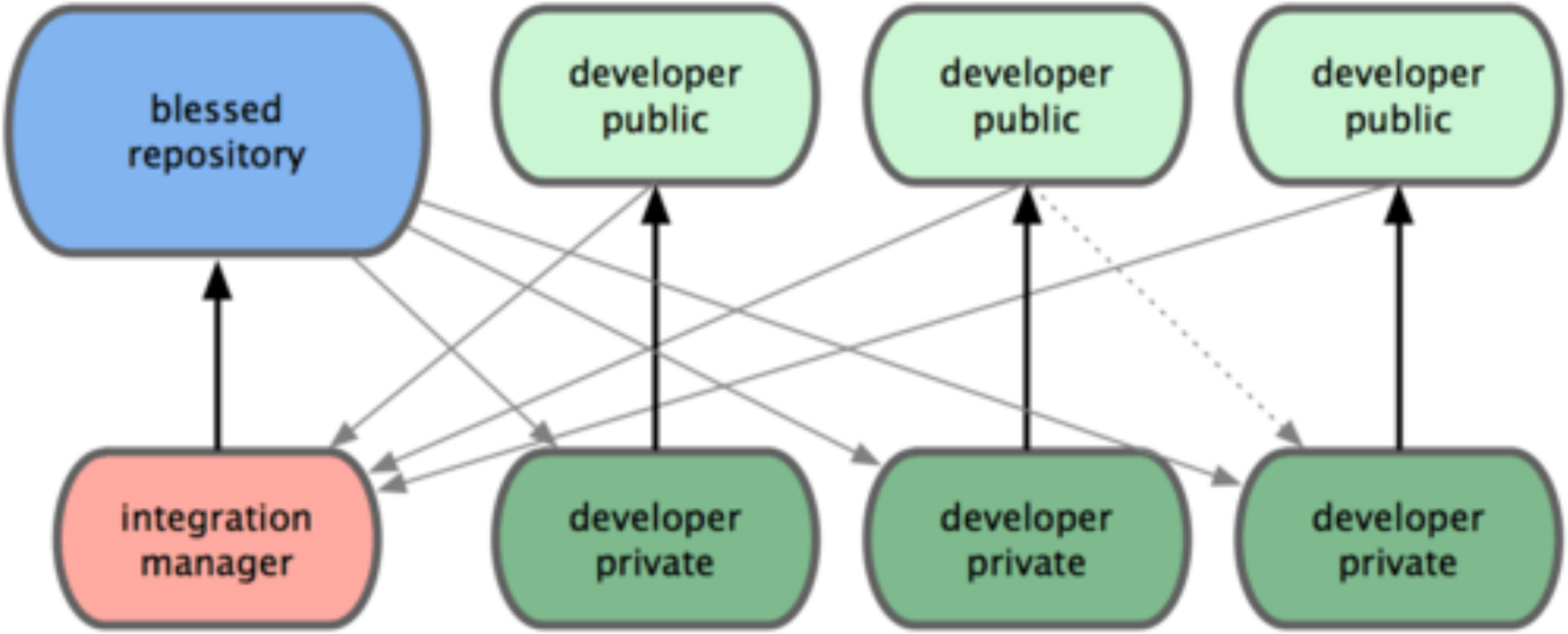
Workflows



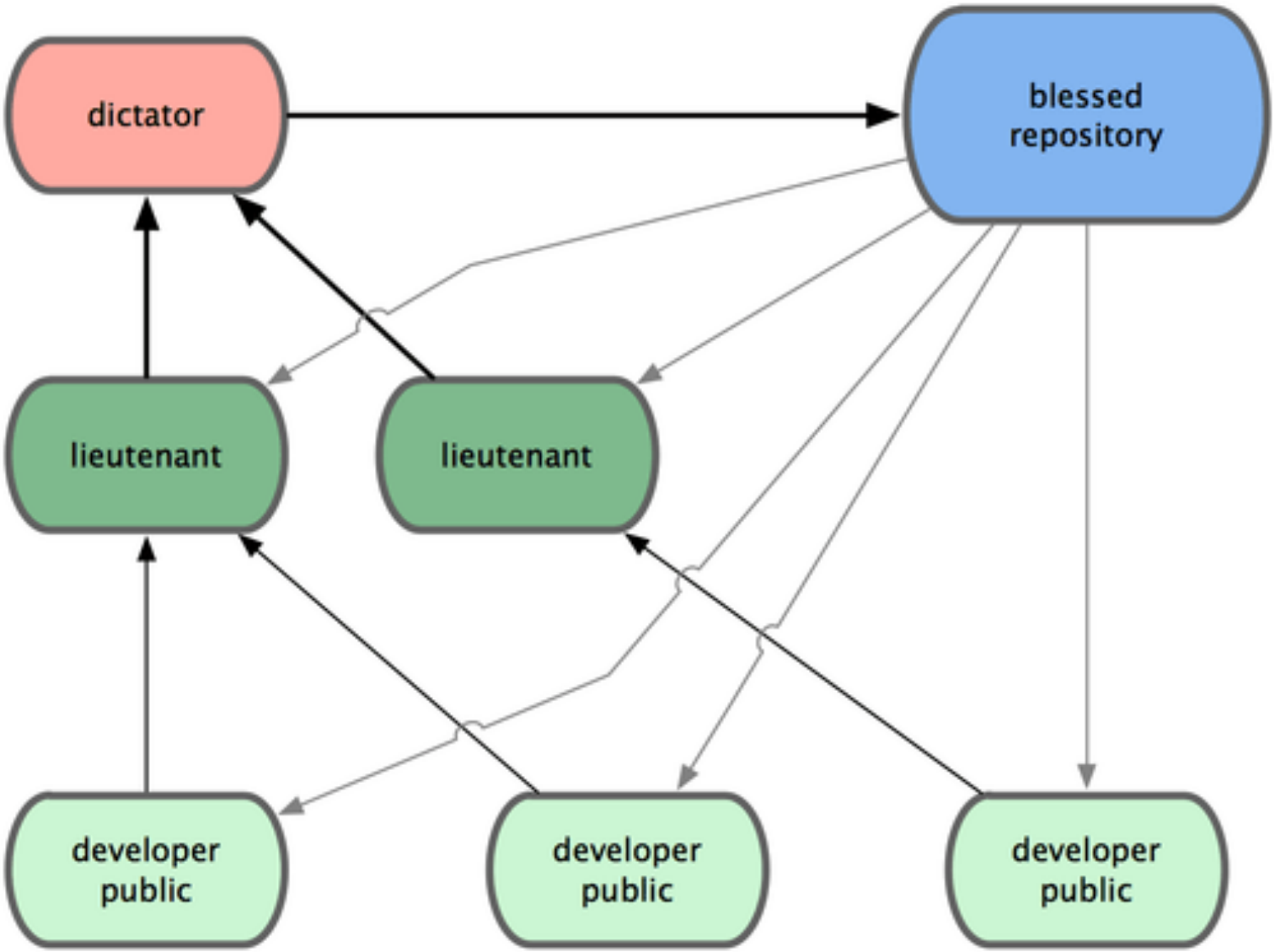
http://git-scm.com/book/en/v1/Distributed-Git-Distributed-Workflows



http://git-scm.com/book/en/v1/Distributed-Git-Distributed-Workflows



http://git-scm.com/book/en/v1/Distributed-Git-Distributed-Workflows



http://git-scm.com/book/en/v1/Distributed-Git-Distributed-Workflows

Cool things that I haven’t covered

- n-way merge (Also, the Cthulhu merge)
  `http://marc.info/?l=linux-kernel&m=139033182525831`
- pull requests
- bisect
- stash
- tags (with signatures and annotations)
- filter-branch
- reset
- hooks
- aliases