

# Git

Ben Lynn

---

Git

Ben Lynn

---



	Git	19
		20
		20
		21
		21
		22
7.	Git	23
		23
		23
		23
	Git	24
	HEAD	24
	HEAD	24
	Git	25
		25
		26
8.		27
		27
		27
		27
		27
	Git	28
		28
	Blob	28
	Tree	28
	Commit	29
		30
9.	A: Git	31
	SHA1	31
	Windows	31
		31
		31
		31
		31
		32
		32
		32
		32
		32
		33
10.	B:	34

---

Git [<http://git.or.cz/>]

Arthur C. Clarke  
Git

Git

Git

- [[/~ blynn/gitmagic/intl/zh\\_cn/](http://blynn/gitmagic/intl/zh_cn/)]: [[/~ blynn/gitmagic/intl/zh\\_tw/](http://blynn/gitmagic/intl/zh_tw/)] + `cconv -f UTF8-CN -t UTF8-TW +`
- [[/~ blynn/gitmagic/intl/fr/](http://blynn/gitmagic/intl/fr/)]: Alexandre Garel itaapy [<http://tutoriels.itaapy.com/>]
- [[/~ blynn/gitmagic/intl/de/](http://blynn/gitmagic/intl/de/)]: Benjamin Bellee Armin Stebich Armin [<http://gitmagic.lordofbikes.de/>]
- [[http://www.slideshare.net/slide\\_user/magia-git](http://www.slideshare.net/slide_user/magia-git)]: Leonardo Siqueira Rodrigues [ODT [[http://www.slideshare.net/slide\\_user/magia-git-verso-odt](http://www.slideshare.net/slide_user/magia-git-verso-odt)]]
- [[/~ blynn/gitmagic/intl/ru/](http://blynn/gitmagic/intl/ru/)]: Tikhon Tarnavsky, Mikhail Dymkov,
- [[/~ blynn/gitmagic/intl/es/](http://blynn/gitmagic/intl/es/)]: Rodrigo Toledo Ariset Llerena Tapia
- [[/~ blynn/gitmagic/intl/vi/](http://blynn/gitmagic/intl/vi/)]: Trần Ng c Quân; [<http://vnwildman.users.sourceforge.net/gitmagic.html>].
- [[book.html](#)]: HTML CSS
- PDF [[book.pdf](#)]: .
- Debian [<http://packages.debian.org/gitmagic>] Ubuntu [<http://packages.ubuntu.com/gitmagic>]; [<http://csdcf.stanford.edu/status/>]
- [Amazon.com [<http://www.amazon.com/Git-Magic-Ben-Lynn/dp/1451523343/>]]: 64 , 15.24cm x 22.86cm,

Dustin Sallings Alberto Bertogli James Cameron Douglas Livingstone Michael Budde Richard Albury Tarmigan Derek Mahar Frode Aannevik Keith Rarick Andy Somerville Ralf Recker Øyvind A. Holm Miklos Vajna Sébastien Hinderer Thomas Miedema Joe Malin Tyler Breisacher

François Marier

Debian

Debian

Daniel Baumann



---

1

[[http://en.wikipedia.org/wiki/Revision\\_control](http://en.wikipedia.org/wiki/Revision_control)]

" "

Speedrun

Git

---

Git

Alice

Bob

Alice

Bob

Alice Bob



---

# 2

Git  
Git

```
$ git init
$ git add .
$ git commit -m "My first backup"
```

```
$ git reset --hard
```

```
$ git commit -a -m "Another backup"
```

git add  
Git

```
$ git add readme.txt Documentation
```

Git

```
$ git rm kludge.h obsolete.c
$ git rm -r incriminating/evidence/
```

Git

git mv mv

```
$ git mv bug.c feature.c
```

/

```
$ git log
```

SHA1 :

```
commit 766f9881690d240ba334153047649b8b8f11c664
Author: Bob <bob@example.com>
Date: Tue Mar 14 01:59:26 2000 -0800
```

Replace printf() with write().

```
commit 82f5ea346a2e651544956a8653c0f58dc151275c
Author: Alice <alice@example.com>
Date: Thu Jan 1 00:00:00 1970 +0000
```

---

Initial commit.

```
$ git reset --hard 766f
```

```
$ git checkout 82f5
```

```
" " branch
```

```
$ git checkout master
```

```
commit      reset      Git      checkout
```

- git reset --hard:
- git checkout:

```
$ git checkout 82f5 some.file another.file
```

```
checkout      checkout      Git      git commit -a  
Git      Git
```

```
$ git checkout :/"My first b"
```

```
$ git checkout master~ 5
```

```
$ git commit -a  
$ git revert 1b6d
```

git log

changelog [<http://en.wikipedia.org/wiki/Changelog>].

---

\$ git log > ChangeLog

Git

\$ git clone git://server/path/to/files

\$ git clone git://git.or.cz/gitmagic.git

clone

git clone

\$ git pull

Git

\$ git init

\$ git add .

\$ git commit -m "First release"

\$ git clone your.computer:/path/to/script

ssh

git daemon

\$ git clone git://your.computer/path/to/script

\$ git commit -a -m "Next release"

\$ git pull

\$ git diff

---

\$ git diff "@ {yesterday}"

\$ git diff 1b6d "master~2"

git apply

\$ git whatchanged --since="2 weeks ago"

```
qgit [http://sourceforge.net/projects/qgit]
tig [http://jonas.nitro.dk/tig/]
web git instaweb
```

```
A B C D      B A
      D
```

D

1. A B

\$ git diff B A | git apply

2. A

\$ git checkout A foo.c bar.h

3. A B

\$ git revert B

Git

---

# 3

checkout

Git

tar rsync

Git

```
$ git clone other.computer:/path/to/files
```

Git

```
$ git commit -a  
$ git pull other.computer:/path/to/files HEAD
```

" "

Git

Git

```
$ git init  
$ git add .  
$ git commit -m "Initial commit"
```

" "

```
$ mkdir proj.git  
$ cd proj.git  
$ git init --bare  
$ touch proj.git/git-daemon-export-ok
```

Git

```
$ git daemon --detach
```

Git

Git

" "

```
$ git push central.server/path/to/proj.git HEAD
```

```
$ git clone central.server/path/to/proj.git
```

```
$ git commit -a
```

---

\$ git pull

\$ git commit -a

\$ git push

pull push SSH

\$ git clone git://central.server/path/to/proj.git

git HTTP git

git touch git SSH `git-daemon-export-ok`  
SSH

`.git`

Git `GIT\_DIR` `--bare`

push pull pull  
" fetch" pull  
shell

Git push pull

\$ git clone git://main.server/path/to/files

---

\$ git pull

?

HEAD 20 SHA1

\$ git clone . /some/new/directory

Git

hardlinking [[http://en.wikipedia.org/wiki/Hard\\_Link](http://en.wikipedia.org/wiki/Hard_Link)],

\$ git pull /the/other/clone HEAD

Git

Git

\$ git init

\$ git add .

\$ git commit -m "Initial commit"

\$ git clone . /some/new/directory

Git

\$ git add .

\$ git commit -m "Sync with everyone else"

\$ git commit -a -m "Description of my changes"

\$ git pull

Subversion,  
Subversion

git svn  
Git Subversion





---

# 4

Git

Git

Git

git branch

" "

```
$ echo "I'm smarter than my boss" > myfile.txt
$ git init
$ git add .
$ git commit -m "Initial commit"
```

Git

```
$ git checkout -b boss
$ echo "My boss is smarter than me" > myfile.txt
$ git commit -a -m "Another commit"
```

```
$ git checkout master
```

```
$ git checkout boss
```

```
$ git commit -a
$ git checkout HEAD~3
```

---

\$ git checkout master

\$ git checkout -b dirty

\$ git checkout dirty

checkout                      Git  
git checkout -b

`1b6d...`

\$ git commit -a  
\$ git checkout -b fixes 1b6d

\$ git commit -a -m "Bug fixed"  
\$ git checkout master

" "

\$ git merge fixes

Git

pull  
" "  
Git  
" "  
HEAD~10

\$ git log HEAD^2

\$ git diff HEAD^

---

\$ git checkout 1b6d^^2~10 -b ancient

" ancient"

\$ git checkout -b part2

\$ git checkout master

\$

\$ git commit -a

\$ git checkout part2

\$ git merge master

\$ git checkout master

\$ submit files

\$ git merge part2

\$ git branch -d part2 " part2"

7

\$ git branch -m master part2

" master"

" part2"

\$ git branch master HEAD~7

" master"

master

part2

master

part2

\$ git checkout HEAD~7 -b master

\$ git branch sanitized

\$ git checkout -b medley

---

\$ git checkout sanitized  
\$ git cherry-pick medley^^

" medley"

" sanitized"

\$ git branch

" master"

" master"

-d -m

git help branch

" master"

" master"

Git

\$ git stash

\$ git stash apply

Git

git help slash

Git

cd

Git

---

# 5

Git

Git

Git

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

```
git add
```

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

```
$ git rebase -i HEAD~10
```

```
10
```

```
$EDITOR
```

```
pick 5c6eb73 Added repo.or.cz link  
pick a311a64 Reordered analogies in "Work How You Want"  
pick 100834f Added push target to Makefile
```

- 
- 
- pick
  - edit
  - reword
  - squash
  - fixup

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

---

\$ git rebase --continue

rebase

Git

git rebase

--onto

git help rebase

\$ git filter-branch --tree-filter 'rm top/secret/file' HEAD

git help filter-branch  
branch

filter-

+ .git/refs/original+

filter-branch

filter-branch

Git

Git

git fast-import

Git

/tmp/history

```
commit refs/heads/master
committer Alice <alice@example.com> Thu, 01 Jan 1970 00:00:00 +0000
data <<EOT
Initial commit.
EOT
```

```
M 100644 inline hello.c
data <<EOT
include <stdio.h>
```

```
int main() {
printf("Hello, world!\n");
```

---

```
    return 0;
}
EOT
```

```
commit refs/heads/master
committer Bob <bob@example.com> Tue, 14 Mar 2000 01:59:26 -0800
data <<EOT
Replace printf() with write().
EOT
```

```
M 100644 inline hello.c
data <<EOT
include <unistd.h>
```

```
int main() {
    write(1, "Hello, world!\n", 14);
    return 0;
}
EOT
```

Git

```
$ mkdir project; cd project; git init
$ git fast-import --date-format=rfc2822 < /tmp/history
```

checkout

```
$ git checkout master .
```

git fast-export

git fast-import

Git

```
$ git bisect start
$ git bisect bad HEAD
$ git bisect good 1b6d
```

Git

```
$ git bisect bad
```

"bad" "good" Git

```
$ git bisect reset
```

```
$ git bisect run my_script
```

---

Git

0

125

1 127

bisect

:

bisects,

bisect

,

Git

"blame"

\$ git blame bug.c

Git

Git

Git



---

# 6 Git

pull      Git  
          \*clone\*      Git

Git      Git

git log      Git

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

global

## Git SSH, HTTP

ssh      Git      HTTP      Git

Git      Git

```
$ GIT_DIR=proj.git git init
$ cd proj.git
$ git --bare update-server-info
$ cp hooks/post-update.sample hooks/post-update
```

Git

```
$ chmod a+x hooks/post-update
```

SSH

```
$ git push web.server:/path/to/proj.git master
```

```
$ git clone http://web.server/proj.git
```

## Git

git fast-export      git fast-import      git

bundle      git

"      "

```
$ git bundle create somefile HEAD
```

OCR      somefile      xxd

\$ git pull somefile

somefile

git

' ^1b6d...! '

\$ git bundle create somefile HEAD ^1b6d

\$ git tag -f lastbundle HEAD

\$ git bundle create newbundle HEAD ^lastbundle

Git

\$ git diff 1b6d > my.patch

Git

\$ git apply < my.patch

\$ git format-patch 1b6d

git-send-email

\$ git format-patch 1b6d..HEAD^^

\$ git am < email.txt

mbox

git push

git pull

URL

Git

\$ git config --list

remote.origin.url    URL    " origin"                    " master"

URL

\$ git config remote.origin.url git://new.url/proj.git

branch.master.merge    git pull

pull

branch.master.remote

\$ git pull git://example.com/other.git master

push    pull

Git

Git

\$ git branch -r

origin/HEAD  
origin/master  
origin/experimental

HEAD

Git

SHA1

\$ git diff origin/HEAD

' `experimental` '

\$ git log origin/experimental

\$ git remote add other git://example.com/some\_repo.git

\$ git pull other some\_branch

\$ git diff origin/experimental^ other/some\_branch~ 5

pull

\$ git fetch            Fetch from origin, the default.

\$ git fetch other    Fetch from the second programmer.

Git

pull

fetch

merge

pull

git help remote

Git

Git

Linus Torvalds

[<http://torvalds-family.blogspot.com/2009/06/happiness-is-warm-scm.html>]

Git

Git

Git

---

# 7 Git

git help

Git

\$ git archive --format=tar --prefix=proj-1.2.3/ HEAD

Git

\$ git add .  
\$ git add -u

Git

`git commit -a`

add

git help ignore

\$ git ls-files -d -m -o -z | xargs -O git update-index --add --remove

-z -O

-x -X

\$ git add -p

" y"      Git  
" n"      " n"

" "

\$ git commit

-a

Git

git add -i

git commit --interactive

# Git

```

Git
    commit -a
    git add -a
  
```

## HEAD

```
HEAD                                     Git
```

```
$ git reset HEAD~3
```

```
HEAD                                     Git
```

```
Head SHA1
```

```
$ git reset 1b6d
```

```
ORIG_HEAD                               Git      Head
```

```
$ git reset ORIG_HEAD
```

## HEAD

```
ORIG_HEAD
```

```
Git                                     Git
```

```
.git/objects
```

```
Git                                     ".git/logs"
HEAD
```

The `reflog` command provides a friendly interface to these log files. Try

```
reflog
```

```
$ git reflog
```

```
reflog
```

```
$ git checkout "@{10 minutes ago}"
```

```
$ git checkout "@ {5}"
```

```
git help rev-parse ' `Specifying Revisions` '
```

```
$ git config gc.pruneexpire "30 days"
```

```
30 git gc
```

```
git gc
```

```
$ git config gc.auto 0
```

```
git gc
```

## Git

UNIX

Git

Web  
Git

Git

Git alias

```
$ git config --global alias.co checkout
```

```
$ git config --global --get-regexp alias
```

```
alias.co checkout
```

```
$ git co foo " git checkout foo"
```

```
$ git symbolic-ref HEAD
```

```
" refs/heads/"
```

```
$ git symbolic-ref HEAD 2> /dev/null | cut -b 12-
```

contrib

Git

Debian Ubuntu

/usr/share/doc/git-core/contrib

workdir/git-new-workdir

```
$ git-new-workdir an/existing/repo new/directory
```

Git

\*Checkout\*

checkout

```
$ git checkout -f HEAD^
```

Reset:

```
$ git reset --hard 1b6d
```

Branch:

```
$ git branch -D dead_branch  instead of -d
```

```
$ git branch -M source target  -m
```

```
checkout .git/logs " HEAD ".git
```

Clean: Git

```
$ git clean -f -d
```

git add

- -

```
$ cd .git/hooks
```

```
$ cp pre-commit.sample pre-commit  Git  chmod +x
```

Git

pre-commit

```
if git ls-files -o | grep '\.txt$'; then
  echo FAIL! Untracked .txt files.
  exit 1
fi
```

git

git help hooks

post-update

```
post-update  HTTP Git  Git  head  HTTP
```



---

# 8

Git  
[<http://www.kernel.org/pub/software/scm/git/docs/user-manual.html>]

Git

Git

check out

Git

.git`

Git

.git`

SHA1

160 ID

ID

SHA1

" "

Git

ID

Git  
ID

`.git/objects`

Git

Git

git mv

git add

git rm

Git

Git

" index"

Git

Git

Git

---

Git

# Git

Linux

[<http://lkml.org/lkml/2005/4/6/121>]

Git

Git

"

"

.git/objects`

.git/`

Git

`git/objects`

3

" blob"

" tree"

" commit"

# Blob

```
$ echo sweet > YOUR_FILENAME
```

```
$ git init
```

```
$ git add .
```

```
$ find .git/objects -type f
```

```
    .git/objects/aa/823728ea7d592acc69b36875a482cdf3fd5c8d
```

SHA1

```
"blob" SP "6" NUL "sweet" LF
```

```
    aa823728ea7d592acc69b36875a482cdf3fd5c8d
```

SP

NUL

O

```
    LF
```

```
$ printf "blob 6\000sweet\n" | sha1sum
```

```
Git
```

```
"
```

```
"
```

```
" blob
```

```
"
```

ID

```
" blob 6"
```

blob

```
.git/objects
```

Git

```
    .git/objects
```

zlib

```
    zpipe -d [http://www.zlib.net/zpipe.c]
```

```
$ git cat-file -p aa823728ea7d592acc69b36875a482cdf3fd5c8d
```

# Tree

Git

---

```

$ git commit
$ find .git/objects -type f
    3
    ' `rose' '

$ git filter-branch --tree-filter 'mv YOUR_FILENAME rose'
$ find .git/objects -type f
    .git/objects/05/b217bb859794d08bb9e4f7f04cbda4b207fbe9
    SHA1

"tree" SP "32" NUL "100644 rose" NUL Oxaa823728ea7d592acc69b36875a482cdf3fd5c8d

$ echo 05b217bb859794d08bb9e4f7f04cbda4b207fbe9 | git cat-file --batch
    zpipe

$ zpipe -d < .git/objects/05/b217bb859794d08bb9e4f7f04cbda4b207fbe9 | sha1sum

    " tree"
    100644      " rose"      blob
" rose"
    tree

$ rm -r .git/refs/original
$ git reflog expire --expire=now --all
$ git prune

    .git      Git
    git update-ref -d

refs/original

```

## Commit

```

" commit"

$ git commit --amend -m Shakespeare
$ git filter-branch --env-filter 'export
    GIT_AUTHOR_DATE="Fri 13 Feb 2009 15:31:30 -0800"
    GIT_AUTHOR_NAME="Alice"
    GIT_AUTHOR_EMAIL="alice@example.com"
    GIT_COMMITTER_DATE="Fri, 13 Feb 2009 15:31:30 -0800"
    GIT_COMMITTER_NAME="Bob"
    GIT_COMMITTER_EMAIL="bob@example.com"' Rig timestamps and authors.
$ find .git/objects -type f
    .git/objects/49/993fe130c4b3bf24857a15d7969c396b7bc187
    SHA1

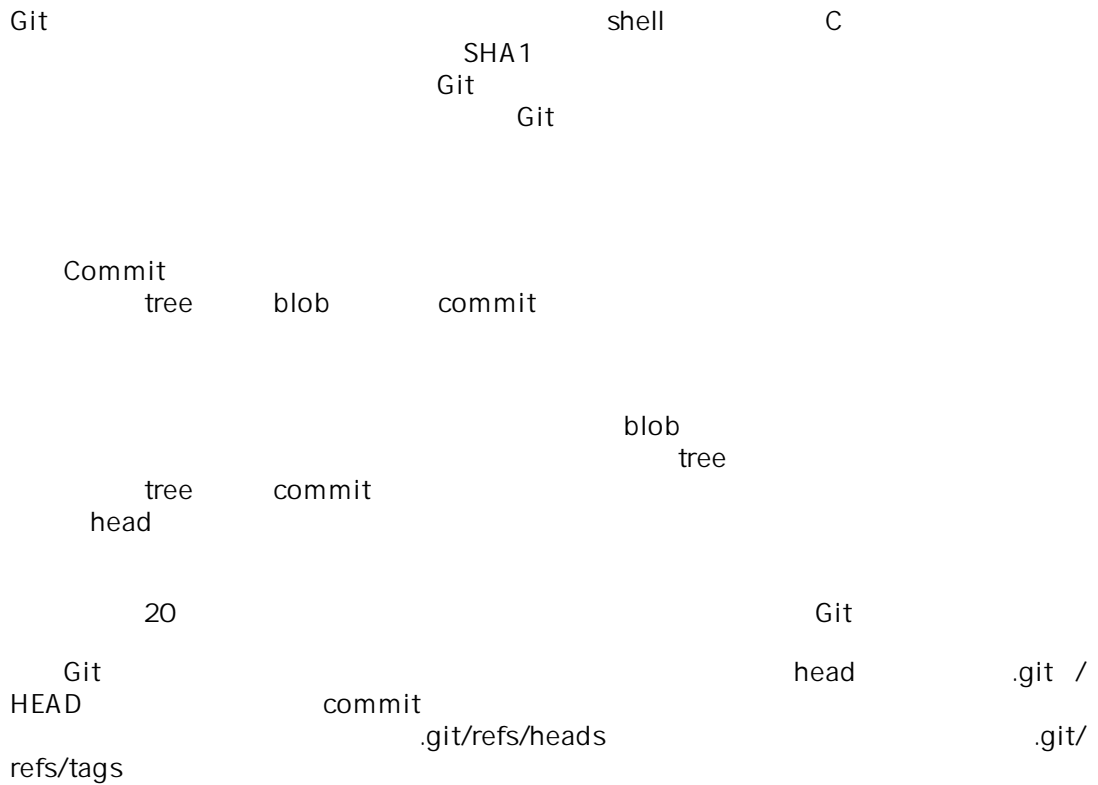
"commit 158" NUL

```

---

"tree 05b217bb859794d08bb9e4f7f04cbda4b207fbe9" LF  
"author Alice <alice@example.com> 1234567890 -0800" LF  
"committer Bob <bob@example.com> 1234567890 -0800" LF  
LF  
"Shakespeare" LF

zpipe cat-file



---

# 9

# A: Git

Git

## SHA1

SHA1

PC

Git

SHA1

Git

## Windows

Git Windows

- Cygwin [<http://cygwin.com/>] , Windows Linux  
Windows [<http://cygwin.com/packages/git/>].
- MSys Git [<http://code.google.com/p/msysgit/>]

Git

Git

Git

git submodule

1.

2.

Git

Git

``cp -a``

Git  
delta

delta

``git checkout``

The initial cost is worth paying in the long run, as most future operations will then be fast and offline. However, in some situations, it may be preferable to create a shallow

clone with the `--depth` option. This is much faster, but the resulting clone has reduced functionality.



Git

HEAD

20

Git

git log

A        B        " A..B"    " A..B"  
git help diff    git help rev-parse

---

# 10

# B:

HTML PDF

language-subtag-registry] IETF tag [http://www.iana.org/assignments/  
W3C  
[http://www.w3.org/International/articles/language-tags/Overview.en.php]  
" en" " ja" " zh-Hant"  
" en" txt

[http://en.wikipedia.org/wiki/Klingon\_language]

```
$ git clone git://repo.or.cz/gitmagic.git
$ cd gitmagic
$ mkdir tlh "tlh" IETF
$ cd tlh
$ cp ../en/intro.txt .
$ edit intro.txt
```

Makefile

TRANSLATIONS

```
$ make tlh
$ firefox book-tlh/index.html
```

GitHub.com

fork

" gitmatic"

Google Docs