免費電子書

學習
Git

Free unaffiliated eBook created from
Stack Overflow contributors.

#git

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: git

It is an unofficial and free Git ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Git.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: Git

Git“”。 。 “Git”Github。

Git“”。 。 Git。

Git3“”

-
-
- Git

GitLinux。 。

CVSSubversion。 。 。 “”。 “”。

2.13	2017510
2.12	2017224
2.11.1	201722
2.11	20161129
2.10.2	20161028
2.10	201692
2.9	2016613
2.8	2016328
2.7	2015104
2.6	2015928
2.5	2015727
2.4	2015430
2.3	201525
2.2	
2.1	2014816
2.0	2014528
1.9	2014214

1.8.3	2013524
1.8	20121021
1.7.10	2012-04-06
1.7	2010-02-13
1.6.5	2009-10-10
1.6.3	2009-05-07
1.6	2008-08-17
1.5.3	2007-09-02
1.5	2007-02-14
1.4	2006-06-10
1.3	2006-04-18
1.2	2006-02-12
1.1	2006-01-08
1.0	2005-12-21
0.99	2005-07-11

Examples

Git

```
git --version
```

UNIX

```
which git
```

Git。 Git 。

Git 。 。

GitGit

```
git init
```

`.git` Git。

Git;

```
git status
```

;Git

```
git add <file/directory name #1> <file/directory name #2> < ... >
```

```
git add .
```

“”。

`.gitignore`add`.gitignore` 。

```
git commit -m "Initial commit"
```

。 。 。

`-m`。

`git remote add`。

`git remote add`

1. `origin`
2. URL`https://<your-git-service-address>/user/repo.git`

```
    git remote add origin https://<your-git-service-address>/owner/repository.git
```

git/。

`git clone`Git。

GitHub

```
cd <path where you'd like the clone to create a directory>
git clone https://github.com/username/projectname.git
```

BitBucket

```
cd <path where you'd like the clone to create a directory>
git clone https://yourusername@bitbucket.org/username/projectname.git
```

`projectname`Git。 `.git`。

`MyFolder`

```
git clone https://github.com/username/projectname.git MyFolder
```

```
git clone https://github.com/username/projectname.git .
```

1. 。
2. `ssh`

   ```
   git clone git@github.com:username/projectname.git
   ```

`https` `ssh`。 GitHub `https` `ssh` 。

forkGithubfork。

```
$ git remote -v
origin    https://github.com/myusername/repo.git (fetch)
origin    https://github.com/myusername/repo.git (push)
upstream  # this line may or may not be here
```

`upstream` GitURL

```
$ git remote set-url upstream https://github.com/projectusername/repo.git
```

/

```
$ git remote add upstream https://github.com/projectusername/repo.git
$ git remote add dave https://github.com/dave/repo.git
```

。

`.git`。 。

```
git init --bare /path/to/repo.git
```

```
git remote add origin ssh://username@server:/path/to/repo.git
```

`ssh:`。

```
git push --set-upstream origin master
```

`--set-upstream` `-u` Git`git pull` 。

```
You need to set who you are *before* creating any commit.  That will allow commits to have the
right author name and email associated to them.
```

GitHubBitBucketGitLab

`git config --global`

`.gitconfig$HOME/.gitconfig`Windows `%USERPROFILE%\.gitconfig`。

```
git config --global user.name "Your Name"
git config --global user.email mail@example.com
```

repo`git config`。

`$GIT_DIR/config`。 `/path/to/your/repo/.git/config`。

```
cd /path/to/my/repo
git config user.name "Your Login At Work"
git config user.email mail_at_work@example.com
```

。

......

-
  ```
  git config --global --remove-section user.name
  git config --global --remove-section user.email
  ```

2.8

- git

  ```
    git config --global user.useConfigOnly true
  ```

`user.nameuser.email`

```
no name was given and auto-detection is disabled
no email was given and auto-detection is disabled
```

git - - `--helphelp`。

`git diff`

```
git diff --help
git help diff
```

`status`

```
git status --help
git help status
```

`-h`

```
git checkout -h
```

GitSSH

WindowsGit Bash 。 **MacLinux**。

SSHSSH。

`~/.ssh`

```
$ ls -al ~/.ssh
# Lists all the files in your ~/.ssh directory
```

SSH。

```
id_dsa.pub
id_ecdsa.pub
id_ed25519.pub
id_rsa.pub
```

BitbucketGitHub`id_*.pub`。

```
$ ssh-keygen
```

EnterReturn。 。

SSHssh-agent。 ssh-agent

```
$ eval "$(ssh-agent -s)"
```

SSHssh-agent。 `id_rsa`

```
$ ssh-add ~/.ssh/id_rsa
```

HTTPSSSH

```
$ git remote set-url origin ssh://git@bitbucket.server.com:7999/projects/your_project.git
```

SSH

```
$ git clone ssh://git@bitbucket.server.com:7999/projects/your_project.git
```

Git

Git。 。 ;。

Git。 GitUI。 Linux;backport。

GitGitcurlzlibopensslexpatlibiconv。 yumFedoraapt-getDebian

```
$ yum install curl-devel expat-devel gettext-devel \
  openssl-devel zlib-devel
```

```
$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \
  libz-dev libssl-dev
```

Git

http://git-scm.com/download

```
$ tar -zxf git-1.7.2.2.tar.gz
$ cd git-1.7.2.2
$ make prefix=/usr/local all
$ sudo make prefix=/usr/local install
```

GitGit

```
$ git clone git://git.kernel.org/pub/scm/git/git.git
```

Linux

LinuxGit。 Fedorayum

```
$ yum install git
```

DebianUbuntuapt-get

```
$ apt-get install git
```

Mac

MacGit。 GitSourceForge。

http://sourceforge.net/projects/git-osx-installer/

1-7。 Git OS X。 MacPorts http://www.macports.orgGit。 MacPortsGit via

```
$ sudo port install git +svn +doc +bash_completion +gitweb
```

+ svnGitSubversion8。

Homebrew http://brew.sh/Git。 HomebrewGit via

```
$ brew install git
```

Windows

WindowsGit。 msysGit。 GitHubinstaller exe

```
http://msysgit.github.io
```

SSHGUI。

*Windows*GitmsysGit shellUnix。 Windows shell /^ Windows。

Git https://riptutorial.com/zh-TW/git/topic/218/git

2: .mailmap

- <primary@example.org> <alias@example.org>
- <primary@example.org>
- 'Other <alias2@example.org>'。 <primary@example.org> <alias1@example.org><alias2@example.org>

`.mailmap`。 `.git`。

`git shortloggit log --use-mailmap` 。 /。

git hooks 。

Examples

。

。

`git shortlog -sn`

```
Patrick Rothfuss 871
Elizabeth Moon 762
E. Moon 184
Rothfuss, Patrick 90
```

`.mailmap`/。

。

```
Patrick Rothfuss <fussy@kingkiller.com> Rothfuss, Patrick <fussy@kingkiller.com>
Elizabeth Moon <emoon@marines.mil> E. Moon <emoon@scifi.org>
```

`git shortlog -sn`

```
Patrick Rothfuss 961
Elizabeth Moon 946
```

.mailmap https://riptutorial.com/zh-TW/git/topic/1270/-mailmap-

3: Bash UbuntuGit

bashgit。 git。 。

Examples

PS1

PS11.Linux / UNIX shell。 bashPS1。 bashPS1/ .bash_profilePS1。

git

/ .bash_profile

```
git_branch() {
 git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/ (\1)/'
}
export PS1="\u@\h \[\033[32m\]\w\[\033[33m\]\$(git_branch)\[\033[00m\] $ "
```

git_branch。 git repo。

Bash UbuntuGit https://riptutorial.com/zh-TW/git/topic/8320/bash-ubuntugit

4: DIFF

blob。

Examples

```
git diff-tree --no-commit-id --name-only -r COMMIT_ID
```

```
git diff-tree [--stdin] [-m] [-c] [--cc] [-s] [-v] [--pretty] [-t] [-r] [--root] [<common-
diff-options>] <tree-ish> [<tree-ish>] [<path>...]
```

-r	diff
-	/ dev / null

-z	diff-rawNUL。
-p	。
-u	-p。
--patch -	。
--stat	diffstatpatch。
--numstat	diffstat。
--patch--STAT	diffstat。
--name	。
--name	。
--full	。
--abbrev = <N>	diff-tree headerdiff-raw。
-R	。
-B	。
-M	。
-C	。

--find-	
-l <N>	。
-O	。
-S	filepair。
--pickaxe	-S。
-a --text	。

DIFF https://riptutorial.com/zh-TW/git/topic/10937/diff

5: Git Clean

- `git clean [-d] [-f] [-i] [-n] [-q] [-e <pattern>] [-x | -X] [--] <path>`

-d	。 Git。 -f。
-f - force	Git。 `requireForce`falsegit clean-f-n-i。 -fGit.git。
-i -	。
-n - dr-run	。
-q -	。

Examples

```
git clean -fX
```

。

```
git clean -Xn
```

。

```
git clean -fd
```

。 。

```
git clean -dn
```

。

```
git clean -f
```

。

```
git clean -i
```

```
Would remove the following items:
  folder/file1.py
  folder/file2.py
*** Commands ***
    1: clean        2: filter by pattern        3: select by numbers        4: ask each
    5: quit        6: help
```

```
What now>
```

iX d◦

Git Clean https://riptutorial.com/zh-TW/git/topic/1254/git-clean

6: Git Diff

- `git diff [options] [<commit>] [--] [<path>…]`
- `git diff [options] --cached [<commit>] [--] [<path>…]`
- `git diff [options] <commit> <commit> [--] [<path>…]`
- `git diff [options] <blob> <blob>`
- `git diff [options] [--no-index] [--] <path> <path>`

-p-u - patch	
-s - no-patch	◦ `git show--patch`
-	diff
--diff=	◦ `myers minimal patience histogram`
-	
--name	
--name	MAD
-	◦ `core.whitespace`◦ ◦ ◦ --exit-code
--full	“”blob
--binary	`--full-index` diff`git apply`
-a - text	◦
-	;`--color=always`difflessgit

Examples

```
git diff
```

◦ ◦ `git add` `git add` ◦

`git diff`◦

```
git diff --staged
```

◦

```
git diff --cached
```

`--staged`

```
git status -v
```

`status`。

```
git diff HEAD
```

```
git status -vv
```

。

```
git diff 1234abc..6789def    # old   new
```

3

```
git diff @~3..@    # HEAD -3   HEAD
```

..。

。

meld

```
git difftool -t meld --dir-diff
```

。

```
git difftool -t meld --dir-diff  [COMMIT_A] [COMMIT_B]
```

2。

```
git diff myfile.txt
```

`myfile.txt` 。

```
git diff documentation
```

`documentation/` 。

`HEAD`

```
git diff 27fa75e myfile.txt
```

```
git diff 27fa75e ada9b57 myfile.txt
```

ada9b57my_branchnamemy_branchname my_changed_directory/

```
git diff ada9b57 my_branchname my_changed_directory/
```

word-diff

```
git diff [HEAD|--staged...] --word-diff
```

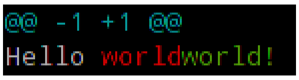
。

```
-Hello world
+Hello world!
```

word-diff

```
Hello [-world-]{+world!+}
```

--word-diff=color--color-words[- -] {+ +}。



```
@@ -1 +1 @@
Hello worldworld!
```

```
git config --global merge.conflictstyle diff3
```

diff3

```
<<<<<<< HEAD
left
=======
right
>>>>>>> master
```

```
<<<<<<< HEAD
first
second
|||||||
first
=======
last
>>>>>>> master
```

。 secondfirstlast

```
last
second
```

```
<<<<<<< HEAD
first
second
=======
last
>>>>>>> master
```

```
git diff HEAD^ HEAD
```

。

Diff UTF-16plist

gitUTF-16os iOSmacOS。

`~/.gitconfig`。

```
[diff "utf16"]
textconv = "iconv -f utf-16 -t utf-8"
```

`iconv`。

`.gitattributes`。 `~/.gitattributes` 。

```
*.strings diff=utf16
```

git `.strings.strings`。

。

plist`.gitconfig`

```
[diff "plist"]
textconv = plutil -convert xml1 -o -
```

`.gitattributes`

```
*.plist diff=plist
```

`neworiginal`

```
git diff original new     # equivalent to original..new
```

`neworiginal`

```
git diff original...new     # equivalent to $(git merge-base original new)..new
```

git diff

git diff original..HEAD

```
git diff branch1..branch2
```

- `git --diff`◦

```
git diff --no-prefix > some_file.patch
```

```
patch -p0 < some_file.patch
```

```
git diff <branch1>..<branch2>
```

```
git diff <commitId1>..<commitId2>
```

```
git diff <branch/commitId>
```

```
git diff --stat <branch/commitId>
```

```
git diff --name-only <commitId>
```

```
git diff --name-only <branchName>
```

```
git diff --name-only <commitId> <folder_path>
```

Git Diff https://riptutorial.com/zh-TW/git/topic/273/git-diff

7: Git GUI

Examples

GitHub

https //desktop.github.com

OS XWindows
GitHub

Git Kraken

https //www.gitkraken.com
60/
LinuxOS XWindows
Axosoft

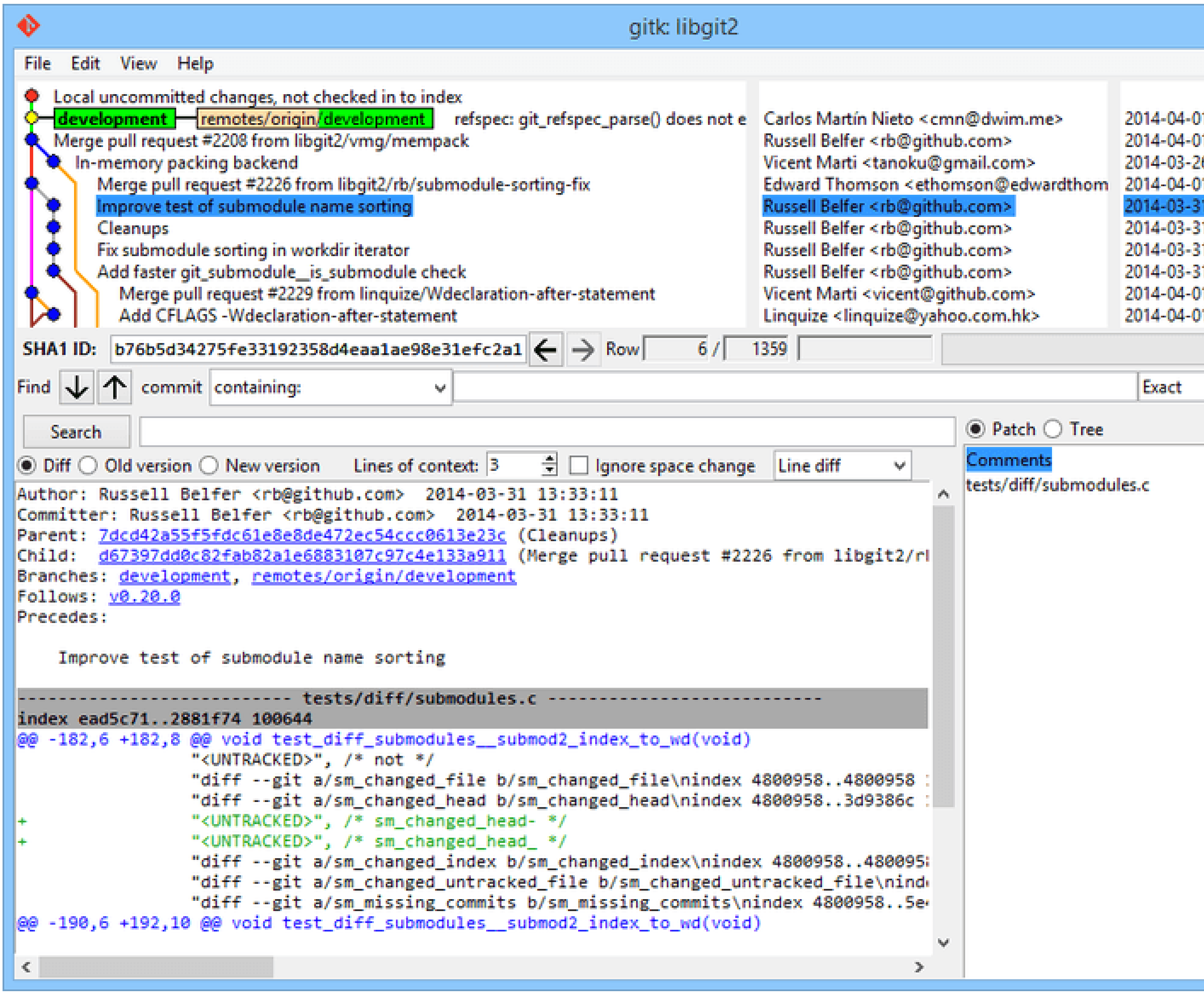
SourceTree

https //www.sourcetreeapp.com

OS XWindows
Atlassian

gitkgit-gui

> Gitgitkgit-gui。
>
> `gitk`。 GUI shellgit loggit grep。 。
>
> Gitk。 cdGit
>
> ```
> $ gitk [git log options]
> ```
>
> Gitkgit log。 `--all`gitkrefHEAD。 Gitk



```
Local uncommitted changes, not checked in to index
development  remotes/origin/development  refspec: git_refspec_parse() does not e   Carlos Martín Nieto <cmn@dwim.me>          2014-04-0
Merge pull request #2208 from libgit2/vmg/mempack                                  Russell Belfer <rb@github.com>             2014-04-0
  In-memory packing backend                                                        Vicent Marti <tanoku@gmail.com>            2014-03-2
    Merge pull request #2226 from libgit2/rb/submodule-sorting-fix                 Edward Thomson <ethomson@edwardthom        2014-04-0
    Improve test of submodule name sorting                                         Russell Belfer <rb@github.com>             2014-03-3
    Cleanups                                                                       Russell Belfer <rb@github.com>             2014-03-3
    Fix submodule sorting in workdir iterator                                      Russell Belfer <rb@github.com>             2014-03-3
    Add faster git_submodule__is_submodule check                                   Russell Belfer <rb@github.com>             2014-03-3
      Merge pull request #2229 from linquize/Wdeclaration-after-statement          Vicent Marti <vicent@github.com>           2014-04-0
      Add CFLAGS -Wdeclaration-after-statement                                     Linquize <linquize@yahoo.com.hk>           2014-04-0
```

SHA1 ID: b76b5d34275fe33192358d4eaa1ae98e31efc2a1 Row 6 / 1359

Find ↓ ↑ commit containing:

● Diff ○ Old version ○ New version Lines of context: 3 ☐ Ignore space change Line diff

● Patch ○ Tree

Comments
tests/diff/submodules.c

```
Author: Russell Belfer <rb@github.com>  2014-03-31 13:33:11
Committer: Russell Belfer <rb@github.com>  2014-03-31 13:33:11
Parent: 7dcd42a55f5fdc61e8e8de472ec54ccc0613e23c (Cleanups)
Child:  d67397dd0c82fab82a1e6883107c97c4e133a911 (Merge pull request #2226 from libgit2/rl
Branches: development, remotes/origin/development
Follows: v0.20.0
Precedes:

    Improve test of submodule name sorting

-------------------------- tests/diff/submodules.c --------------------------
index ead5c71..2881f74 100644
@@ -182,6 +182,8 @@ void test_diff_submodules__submod2_index_to_wd(void)
                 "<UNTRACKED>", /* not */
                 "diff --git a/sm_changed_file b/sm_changed_file\nindex 4800958..4800958 :
                 "diff --git a/sm_changed_head b/sm_changed_head\nindex 4800958..3d9386c :
+                "<UNTRACKED>", /* sm_changed_head- */
+                "<UNTRACKED>", /* sm_changed_head_ */
                 "diff --git a/sm_changed_index b/sm_changed_index\nindex 4800958..480095:
                 "diff --git a/sm_changed_untracked_file b/sm_changed_untracked_file\nind
                 "diff --git a/sm_missing_commits b/sm_missing_commits\nindex 4800958..5e
@@ -190,6 +192,10 @@ void test_diff_submodules__submod2_index_to_wd(void)
```

1-1◦ gitk◦

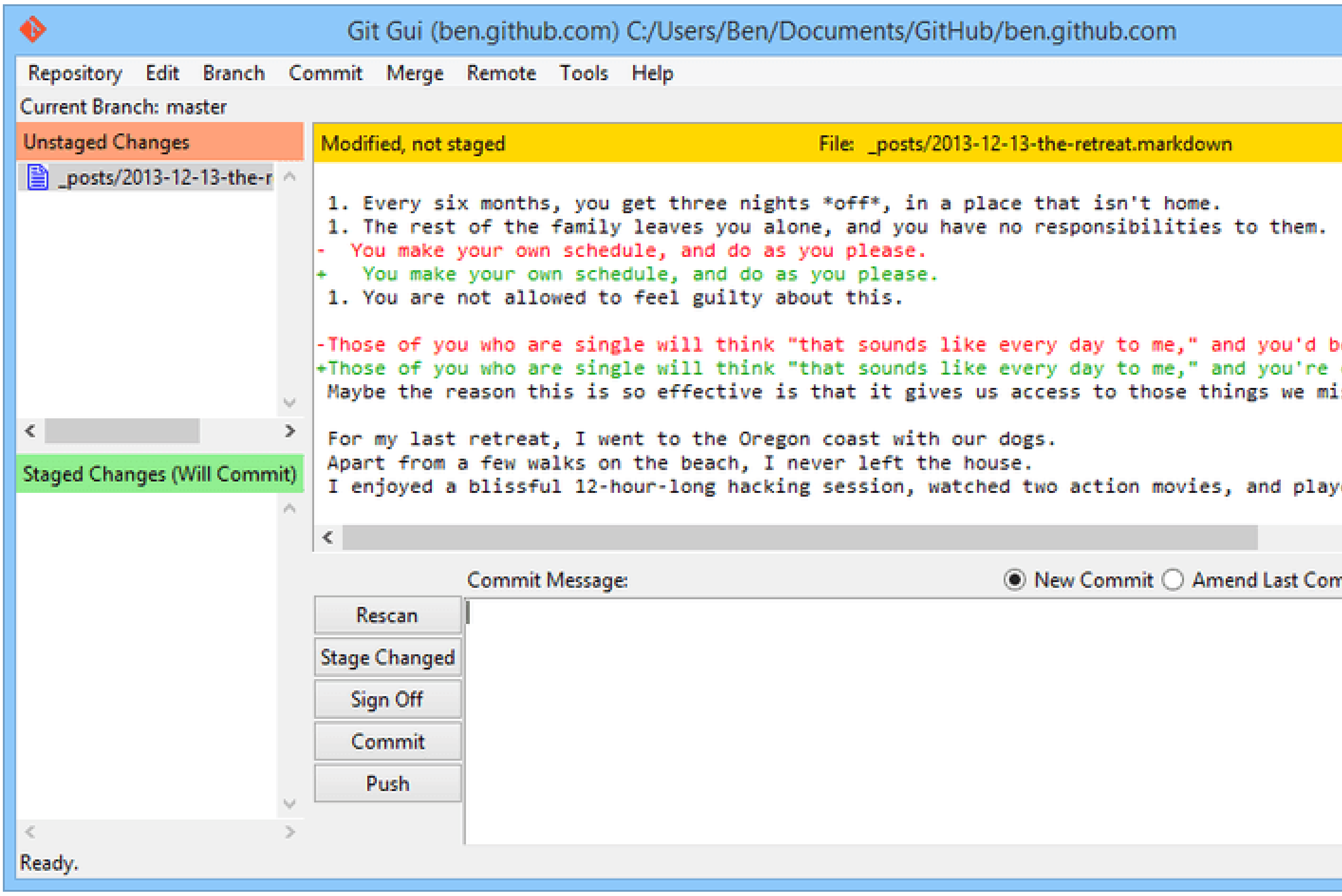
git log --graph;refs◦ HEAD◦ ;◦ ◦

git◦ ◦

git-gui◦

```
$ git gui
```

git-gui◦



```
 1. Every six months, you get three nights *off*, in a place that isn't home.
 1. The rest of the family leaves you alone, and you have no responsibilities to them.
-  You make your own schedule, and do as you please.
+   You make your own schedule, and do as you please.
 1. You are not allowed to feel guilty about this.

-Those of you who are single will think "that sounds like every day to me," and you'd b
+Those of you who are single will think "that sounds like every day to me," and you're
 Maybe the reason this is so effective is that it gives us access to those things we mi

 For my last retreat, I went to the Oregon coast with our dogs.
 Apart from a few walks on the beach, I never left the house.
 I enjoyed a blissful 12-hour-long hacking session, watched two action movies, and play
```

1-2。 git-gui。

;。 。

diff。 。

。 “”git commit。 “Amend”“Staged Changes”。 “”。

gitkgit-gui。 。

https //git-scm.com/book/en/v2/Git-in-Other-Environments-Graphical-Interfaces

SmartGit

http //www.syntevo.com/smartgit/
。 99
LinuxOS XWindows
syntevo

Git

https //gitextensions.github.io

Windows

Git GUI https://riptutorial.com/zh-TW/git/topic/5148/git-gui

8: Git Patch

- git am [--signoff] [--keep] [- [no-] keep-cr] [- [no-] utf8] [--3way] [--interactive] [--committer-date-is -author-date] [--ignore-date] [--ignore-space-change | --ignore-whitespace] [--whitespace = <option>] [-C <n>] [-p <n>] [--directory = <dir>] [--exclude = <path>] [- include = <path>] [--reject] [-q | --quiet] [- [no-] scissors] [-S [<keyid>]] [- patch-format = <format>] [<mbox> | <Maildir>...]
- git am--continue | --skip | --abort

<MBOX> | <Maildir>...	。 。 Maildirs。
-s -	Signed-off-by。
-q - quiet	。 。
-u - utf8	`-ugit mailinfo` 。 UTF-8 `i18n.commitencoding`UTF-8。 `--no-utf8`。
--no-UTF8	-ngit mailinfo。
-3 - 3way	blobblob。
--ignore-date - ignore-space-change - ignore-whitespace--whitespace = <option>-C <n> - p <n> - directory = <dir> - exclude = <path> - include = <path> - repeat	git。
--patch	。 。 `mbox stgit stgit-serieshg` 。
-i -	。
--committer-	。 。
-	。 。
-	。 。
-S [<keyid>] - gpg-sign [= <keyid>]	GPG。
- -r -	。 。
--resolvemsg = <MSG>	`<msg>`。 `--continue--skip`。 `git rebasegit am`。
--abort	。

Examples

。

1. 。
2. `git format-patch <commit-reference>` <commit-reference>。

```
git format-patch HEAD~~
```

2`HEAD~~`

```
0001-hello_world.patch
0002-beginning.patch
```

`git apply some.patch.patch`。 。

```
git am some.patch
```

```
git am *.patch
```

Git Patch https://riptutorial.com/zh-TW/git/topic/4603/git-patch

9: Git Remote

- `git remote [-v | --verbose]`
- `git remote add [-t <branch>] [-m <master>] [-f] [--[no-]tags] [--mirror=<fetch|push>]<name> <url>`
- `git remote rename <old> <new>`
- `git remote remove <name>`
- `git remote set-head <name> (-a | --auto | -d | --delete | <branch>)`
- `git remote set-branches [--add] <name> <branch>...`
- `git remote set-url [--push] <name> <newurl> [<oldurl>]`
- `git remote set-url --add [--push] <name> <newurl>`
- `git remote set-url --delete [--push] <name> <url>`
- `git remote [-v | --verbose] show [-n] <name>...`
- `git remote prune [-n | --dry-run] <name>...`
- `git remote [-v | --verbose] update [-p | --prune] [(<group> | <remote>)...]`
- `git remote show <name>`

-v - verbose	。
-m <master>	headremote<master>
--mirror =	Refsrefs / remotes
--mirror =	`git push`--mirror
--no	`git fetch <name>`
-t <branch>	<branch>
-F	`git fetch <name>`
--tags	`git fetch <name>`
-a - auto	symbolic-refHEADHEAD
-d -	
-	<name>set-branches
-	set-url
-	。
-	<url>。 URL
-	URLSURL
-n	`git ls-remote <name>`
--dry	

-	

Examples

`git remote add`。

Git<url><name>

```
git remote add <name> <url>
```

`git fetch <name><name>/<branch>`。

`<old><new>`。 。

`devdev1`

```
git remote rename dev dev1
```

`<name>`。 。

`dev`

```
git remote rm dev
```

`git remote`。

。

```
$ git remote
premium
premiumPro
origin
```

`--verbose-v`。 URL `pushpull`

```
$ git remote -v
premiumPro    https://github.com/user/CatClickerPro.git (fetch)
premiumPro    https://github.com/user/CatClickerPro.git (push)
premium    https://github.com/user/CatClicker.git (fetch)
premium    https://github.com/user/CatClicker.git (push)
origin    https://github.com/ud/starter.git (fetch)
origin    https://github.com/ud/starter.git (push)
```

GitURL

。 URL

```
git remote set-url
```

2originupstreamurl。

```
git remote -v
origin    https://bitbucket.com/develop/myrepo.git (fetch)
origin    https://bitbucket.com/develop/myrepo.git (push)
```

```
git remote set-url origin https://localserver/develop/myrepo.git
```

```
git remote -v
origin    https://localserver/develop/myrepo.git (fetch)
origin    https://localserver/develop/myrepo.git (push)
```

`git remote show <remote repository alias>`

```
git remote show origin
```

```
remote origin
Fetch URL:  https://localserver/develop/myrepo.git
Push  URL:  https://localserver/develop/myrepo.git
HEAD branch: master
Remote branches:
  master      tracked
Local branches configured for 'git pull':
  master      merges with remote master
Local refs configured for 'git push':
  master      pushes to master      (up to date)
```

Git Remote https://riptutorial.com/zh-TW/git/topic/4071/git-remote

10: Git rerere

`rerere` git。 git。

Examples

rerere

`rerere`

```
$ git config --global rerere.enabled true
```

。

Git rerere https://riptutorial.com/zh-TW/git/topic/9156/git-rerere

11: git send-email

- git send-email [options] <file | directory | rev-list options> ...
- git send-email --dump-aliases

https://git-scm.com/docs/git-send-email

Examples

Gmailgit send-email

Linuxlistserv。 Gmailgit-send。

.gitconfig

```
[sendemail]
    smtpserver = smtp.googlemail.com
    smtpencryption = tls
    smtpserverport = 587
    smtpuser = name@gmail.com
```

Google - > - > - > - >

```
git format-patch HEAD~~~~ --subject-prefix="PATCH <project-name>"
```

listserv

```
git send-email --annotate --to project-developers-list@listserve.example.com 00*.patch
```

2

```
git format-patch -v 2 HEAD~~~~  ......
git send-email --to project-developers-list@listserve.example.com v2-00*.patch
```

--from * Email From - [no-] to * Email To - [no-] cc * Email Cc - [no-] bcc * Email Bcc--subject * Email“Subject” - -in-reply-to *“In-Reply-To” - [no-] xmailer *“X-Mailer”。 - [no-] annotate *。 - *。 --compose-encoding *。 --8bit-encoding *8--transfer-encoding *quoted-printable8bitbase64

ulogd2git-svnMaillingdevel@netfilter.org。 gitshell

```
git format-patch --stat -p --raw --signoff  --subject-prefix="ULOGD PATCH" -o /tmp/ulogd2/ -n
git-svn
git send-email --compose --no-chain-reply-to --to devel@netfilter.org /tmp/ulogd2/
```

/ tmp / ulogd2 /。

```
git config sendemail.chainreplyto false
```

git send-email https://riptutorial.com/zh-TW/git/topic/4821/git-send-email

12: GIT-SVN

SVN

SVN repogit-svnSVN repo。 SVN repo;gitrepo。 SVN。

SHA1

`git svn dcommit`git。 git commitSVNSVN。 git。 git commitSHA1

git-svngitgitSHA1 IDgitSHA1gitSHA1。 SVNsvn。

SHA1/

---

git svn rebase

git svn rebase

```
Checksum mismatch: <path_to_file> <some_kind_of_sha1>
expected: <checksum_number_1>
  got: <checksum_number_2>
```

svngit svn fetchSVN。 SVN

- git log -1 - `<path_to_file>` SVN
- git svn reset `<revision_number>`
- git svn fetch

SVN/

SVN

```
<file_path> was not found in commit <hash>
```

SVN。 SVN

- `git svn fetch --ignore-paths <file_path>`

Examples

SVN

`git svn clone SVN_REPO_ROOT_URL [DEST_FOLDER_PATH] -T TRUNK_REPO_PATH -t TAGS_REPO_PATH -b BRANCHES_REPO_PATH`

SVN

`git svn clone -s SVN_REPO_ROOT_URL [DEST_FOLDER_PATH]` `git svn clone`SVNgit。 SVN。

gitmasterSVNtrunk。

SVN

`git pull`

```
git svn rebase
```

SVN 。

```
git svn fetch
```

SVN。

SVN

```
git svn dcommit
```

gitSVN。 SVNgitSVN`git svn rebase` 。

gitgit repogit

- `git add FILE``git checkout -- FILE``git checkout -- FILE` /`git checkout -- FILE`
- `git commit`。 “”SVN repogit
- `git stash``git stash pop`stashes
- `git reset HEAD --hard`
- `git log`
- `git rebase -i`
- `git branch``git checkout`

git-svn“SubversionGit”Subversiongit。

git//。 。 `git rebase``git rebase` 。

。 。 “”SVN。

git merge“”SVN 。 git merge commitsvn。

git。 git。 SVN。 git-svngit *SVN* 。

`--rmdir`SVN

`git svn dcommit --rmdir`

。

dcommitgit GUISourceTreedefault

```
git config --global svn.rmdir true
```

.gitconfig

```
[svn]
rmdir = true
```

SVNgit

```
git clean -fd
```

SVNagaing SVN

```
git svn mkdirs
```

SVN

```
git clean -fd && git svn mkdirs
```

GIT-SVN https://riptutorial.com/zh-TW/git/topic/2766/git-svn

13: GIT-TFS

Git-tfsGitTeam Foundation Server“TFS”。

TFVSGit-Credential-Manager-for-Windows。 `.git/config`

```
[tfs-remote "default"]
  url = http://tfs.mycompany.co.uk:8080/tfs/DefaultCollection/
  repository = $/My.Project.Name/
  username = me.name
  password = My733TPwd
```

Examples

git-tfs

/My.Project.Name

```
$ git tfs clone http://tfs:8080/tfs/DefaultCollection/ $/My.Project.Name
```

gitgit-tfs

gitTFVS。 git-tfsgit。 TFVS。

```
$ git clone x:/fileshare/git/My.Project.Name.git
$ cd My.Project.Name
$ git tfs bootstrap
$ git tfs pull
```

git-tfsChocolatey

kdiff3。

```
C:\> choco install kdiff3
```

Git。 Unix'NoAutoCrlf'。

```
C:\> choco install git -params '"/GitAndUnixToolsOnPath /NoAutoCrlf"'
```

chocolateygit-tfs。

```
C:\> choco install git-tfs
```

git-tfs

TFVS“”。

```
$ git tfs checkintool
```

。

git-tfs

TFVS。

```
$ git tfs rcheckin
```

。 `git-tfs-force: rcheckin`。

GIT-TFS https://riptutorial.com/zh-TW/git/topic/2660/git-tfs

14: Git

Examples

Git

```
/build
app.js
```

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

Gitapp.js.

“build”“.gitkeep”Git。

```
/build
  .gitkeep
app.js
```

```
git add build/.gitkeep
git commit -m "Keep the build directory around"
```

Gitbuild / .gitkeep。

Git。

Git https://riptutorial.com/zh-TW/git/topic/2680/git

15: Git

Git。 git-log1 。 `<commit>` `<rev>``<revision>` 。

Gitgitrevisions7。

- ⎣⎦ `git describev1.7.4.2-679-g3bee7fb`
- ⎣⎦ `@HEAD`
- ⎣⎦ `@{-<n>}` `@{-1}` `-@{-1}`
- ⎣⎦ `<branchname>@{push}`
- ⎣⎦ `<rev>^@` `<rev>`

- ⎣⎦blob `<rev>:<path>:<n>:<path>`
- ⎣⎦`A..B` `A...B` `B` `^A` `A^1-<n>` - `--since`

Examples

```
$ git show dae86e1950b1277e545cee180551750029cfe735
$ git show dae86e19
```

SHA-140blob。

```
$ git log master    # specify branch
$ git show v1.0     # specify tag
$ git show HEAD     # specify current branch
$ git show origin   # specify default remote-tracking branch for remote 'origin'
```

“master”“next”“maint”“v1.0”“v0.6.3-rc2”remote-'origin''origin / master''HEAD'。

“fix”

```
$ git show heads/fix      # or 'refs/heads/fix', to specify branch
$ git show tags/fix       # or 'refs/tags/fix', to specify tag
```

HEAD

```
$ git show         # equivalent to 'git show HEAD'
```

'HEAD'。 “HEAD”。

Reflog @ { }

```
$ git show @{1}            # uses reflog for current branch
$ git show master@{1}      # uses reflog for branch 'master'
$ git show HEAD@{1}        # uses 'HEAD' reflog
```

refHEAD`@ {1} {15}` n。 `git reflog`reflog`--walk-reflogs` / `-g` `git log` 。

```
$ git reflog
08bb350 HEAD@{0}: reset: moving to HEAD^
4ebf58d HEAD@{1}: commit: gitweb(1): Document query parameters
08bb350 HEAD@{2}: pull: Fast-forward
f34be46 HEAD@{3}: checkout: moving from af40944bda352190f05d22b7cb8fe88beb17f3a7 to master
af40944 HEAD@{4}: checkout: moving from master to v2.6.3

$ git reflog gitweb-docs
4ebf58d gitweb-docs@{0}: branch: Created from master
```

reflogs`ORIG_HEAD` ref`HEAD@{1}` 。

Reflog @ { }

```
$ git show master@{yesterday}
$ git show HEAD@{5 minutes ago}   # or HEAD@{5.minutes.ago}
```

ref`@ {yesterday}` `{1 month 2 weeks 3 days 1 hour 1 second ago}{1979-02-26 18:30:00}` ref。 ;“”。

`git reflog`。

```
$ git reflog HEAD@{now}
08bb350 HEAD@{Sat Jul 23 19:48:13 2016 +0200}: reset: moving to HEAD^
4ebf58d HEAD@{Sat Jul 23 19:39:20 2016 +0200}: commit: gitweb(1): Document query parameters
08bb350 HEAD@{Sat Jul 23 19:26:43 2016 +0200}: pull: Fast-forward
```

/ @{}

```
$ git log @{upstream}..       # what was done locally and not yet published, current branch
$ git show master@{upstream}  # show upstream of branch 'master'
```

branchname`@{upstream}` `<branchname>@{u}` branchname`branch.<name>.remote` `branch.<name>.remote` `branch.<name>.merge` `git branch --set-upstream-to=<branch>` 。 branchname。

```
$ git log --oneline @{u}..
$ git log --oneline ..@{u}
$ git log --oneline --left-right @{u}...  # same as ...@{u}
```

^

```
$ git reset --hard HEAD^             # discard last commit
$ git rebase --interactive HEAD~5    # rebase last 4 commits
```

`^`。 `^<n>`n`<rev>^<rev>^1` 。

`~<n>`n。 `<rev>~3<rev>^^^` 。 `<rev>~<rev>~1` `<rev>^1` `<rev>^` 。

。

`git name-rev`

```
$ git name-rev 33db5f4d9027a10e477ccf054b2c1ab94f74c85a
33db5f4d9027a10e477ccf054b2c1ab94f74c85a tags/v0.99~940
```

`--pretty=oneline` `--oneline`

```
$ git log --pretty=oneline | git name-rev --stdin --name-only
master Sixth batch of topics for 2.10
master~1 Merge branch 'ls/p4-tmp-refs'
master~2 Merge branch 'js/am-call-theirs-theirs-in-fallback-3way'
[...]
master~14^2 sideband.c: small optimization of strbuf usage
master~16^2 connect: read $GIT_SSH_COMMAND from config file
[...]
master~22^2~1 t7810-grep.sh: fix a whitespace inconsistency
master~22^2~2 t7810-grep.sh: fix duplicated test name
```

^ 0 ^ { }

。 “”。

`^` `tag` `commit` `tree` `blob` `v0.99.8^{commit}` `<rev><type>`。 `<rev>^0<rev>^{commit}`。

```
$ git checkout HEAD^0    # equivalent to 'git checkout --detach' in modern Git
```

`^v0.99.8^{}` 。

```
$ git show v1.0
$ git cat-file -p v1.0
$ git replace --edit v1.0
```

```
$ git show v1.0^{}
$ git cat-file -p v1.0^{}
$ git replace --edit v1.0^{}
```

^ {/ }/ /

```
$ git show HEAD^{/fix nasty bug}   # find starting from HEAD
$ git show ':/fix nasty bug'       # find starting from any branch
```

`'` `:` `"` `/` `'`。 ref。 。 `:/^foo` 。 `:/!`。 `:/!-foo:/!!foofoo` 。

`^` `:/<text><rev>^` 。

Git https://riptutorial.com/zh-TW/git/topic/3735/git

16: GitLFS

Git LFSGit。 LFSgit。

LFS;PSDJPEG;3D。 。

LFSGitHub https://github.com/blog/1986-announcing-git-large-file-storage-lfs ;Atlasssiangit-lob 。 。

Examples

LFS

Homebrew。

Brew

```
brew install git-lfs
git lfs install
```

lfs。 git lfs。

Git LFS`.gitignore`。

。

```
git lfs track "*.psd"
```

。

lfs`.gitattributes`。 Github`.gitattributes.gitattributes`。

LFS

LFS `.lfsconfig.lfsconfig`。 `.git/config`LFS。

LFS`.lfsconfig`

```
[lfs]
    fetchexclude = ReallyBigFile.wav
```

GitLFS https://riptutorial.com/zh-TW/git/topic/4136/git-lfs-

17: Git

Git。 。 。 。

Examples

Git`hooks`。 `.git/hooks` 。

`.githooks`。

Git

`git push`。 。

```
 $1 -- Name of the remote to which the push is being done (Ex: origin)
 $2 -- URL to which the push is being done (Ex:
https://<host>:<port>/<username>/<project_name>.git)
```

```
<local_ref> <local_sha1> <remote_ref> <remote_sha1>
```

```
local_ref = refs/heads/master
local_sha1 = 68a07ee4f6af8271dc40caae6cc23f283122ed11
remote_ref = refs/heads/master
remote_sha1 = efd4d512f34b11e3cf5c12433bbedd4b1532716f
```

pre-push.sample`git init`。

```
# This sample shows how to prevent push of commits where the log message starts
# with "WIP" (work in progress).

remote="$1"
url="$2"

z40=0000000000000000000000000000000000000000

while read local_ref local_sha remote_ref remote_sha
do
    if [ "$local_sha" = $z40 ]
    then
        # Handle delete
        :
    else
        if [ "$remote_sha" = $z40 ]
        then
            # New branch, examine all commits
            range="$local_sha"
        else
            # Update to existing branch, examine new commits
            range="$remote_sha..$local_sha"
        fi
        # Check for WIP commit
        commit=`git rev-list -n 1 --grep '^WIP' "$range"`
        if [ -n "$commit" ]
        then
            echo >&2 "Found WIP commit in $local_ref, not pushing"
            exit 1
        fi
    fi
done

exit 0
```

Git https://riptutorial.com/zh-TW/git/topic/8654/git

18: Git

VCS `Gittag。 v1.0 。`

- git[-a | -s | -u <keyid>] [- f] [-m <msg> | -F <file>] <tagname> [<commit> | <object>]
- git tag -d <tagname>
- git tag [-n [<num>]] -l [--contains <commit>] [--contains <commit>] [--points-at <object>] [--column [= <options>] | --no-column] [--create-reflog] [--sort = <key>] [--format = <format>] [- [no-] merged [<commit>]] [<pattern> ...]
- git tag -v [--format = <format>] <tagname> ...

Examples

`git tag`

```
$ git tag
<output follows>
v0.1
v1.3
```

`tags 。`

`searchtags`

```
$ git tag -l "v1.8.5*"
<output follows>
v1.8.5
v1.8.5-rc0
v1.8.5-rc1
v1.8.5-rc2
v1.8.5-rc3
v1.8.5.1
v1.8.5.2
v1.8.5.3
v1.8.5.4
v1.8.5.5
```

GIT

-
  ```
  git tag < tagname >
  ```

  `tag 。`

-
  ```
  git tag tag-name commit-identifier
  ```

  `tag` commit-identifier。

GIT

-
  ```
  git push origin tag-name
  ```

-
  ```
  git push origin --tags
  ```

Git https://riptutorial.com/zh-TW/git/topic/10098/git

19: Git

- git log [<options>] [<revision range>] [[-] <path>]
- git log --pretty = short | git shortlog [<options>]
- git shortlog [<options>] [<revision range>] [[-] <path>]

`-n --numbered`	
`-s` - `--summary`	
`-e` - `--email`	
`--format` [= <format>]	。 <format>`git log--format`。
`-w` [<width> [<indent1> [<indent2>]]]	`width`。 `indent1indent2`。
<>	。 。
[--] <path>	`path`。 “ - ”。

Examples

Git `shortlog`git。

`--summary-s`。

`--numbered-n --numbered`。

```
git shortlog -sn        #Names and Number of commits

git shortlog -sne       #Names along with their email ids and the Number of commits
```

```
git log --pretty=format:%ae \
| gawk -- '{ ++c[$0]; } END { for(cc in c) printf "%5d %s\n",c[cc],cc; }'
```

/。 `John DoeJohnny Doe`。 `.mailmap`。

```
git log --pretty=format:"%ai" | awk '{print " : "$1}' | sort -r | uniq -c
```

```
git log  --pretty=oneline |wc -l
```

```
for k in `git branch -a | sed s/^..//`; do echo -e `git log -1 --pretty=format:"%Cgreen%ci
%Cblue%cr%Creset" $k --`\\t"$k";done | sort
```

```
git ls-tree -r HEAD | sed -Ee 's/^.{53}//' | \
while read filename; do file "$filename"; done | \
grep -E ': .*text' | sed -E -e 's/: .*//' | \
while read filename; do git blame --line-porcelain "$filename"; done | \
sed -n 's/^author //p' | \
sort | uniq -c | sort -rn
```

```
git log --pretty=format:"%Cgreen%ci %Cblue%cn  %Cgreen%cr%Creset %s"
```

1。

`--pretty%`。

Git

git

```
find $HOME -type d -name ".git"
```

`locate`

```
locate .git |grep git$
```

`gnu locatemlocate` git dirs

```
locate -ber \\.git$
```

`git shortlog`

```
git shortlog -s
```

。

`--all`

```
git shortlog -s --all
```

Git https://riptutorial.com/zh-TW/git/topic/4609/git

20: Reflog - git

GitreflogHEADref。 HEAD。 reflog。

ref~30reflog。

Examples

rebase

rebase

```
git rebase --interactive HEAD~20
```

rebase。 `git reflog`

```
aaaaaaa HEAD@{0} rebase -i (finish): returning to refs/head/master
bbbbbbb HEAD@{1} rebase -i (squash): Fix parse error
...
ccccccc HEAD@{n} rebase -i (start): checkout HEAD~20
ddddddd HEAD@{n+1} ...
...
```

`ddddddd HEAD@{n+1}` *rebase*。

```
$ git checkout HEAD@{n+1}
```

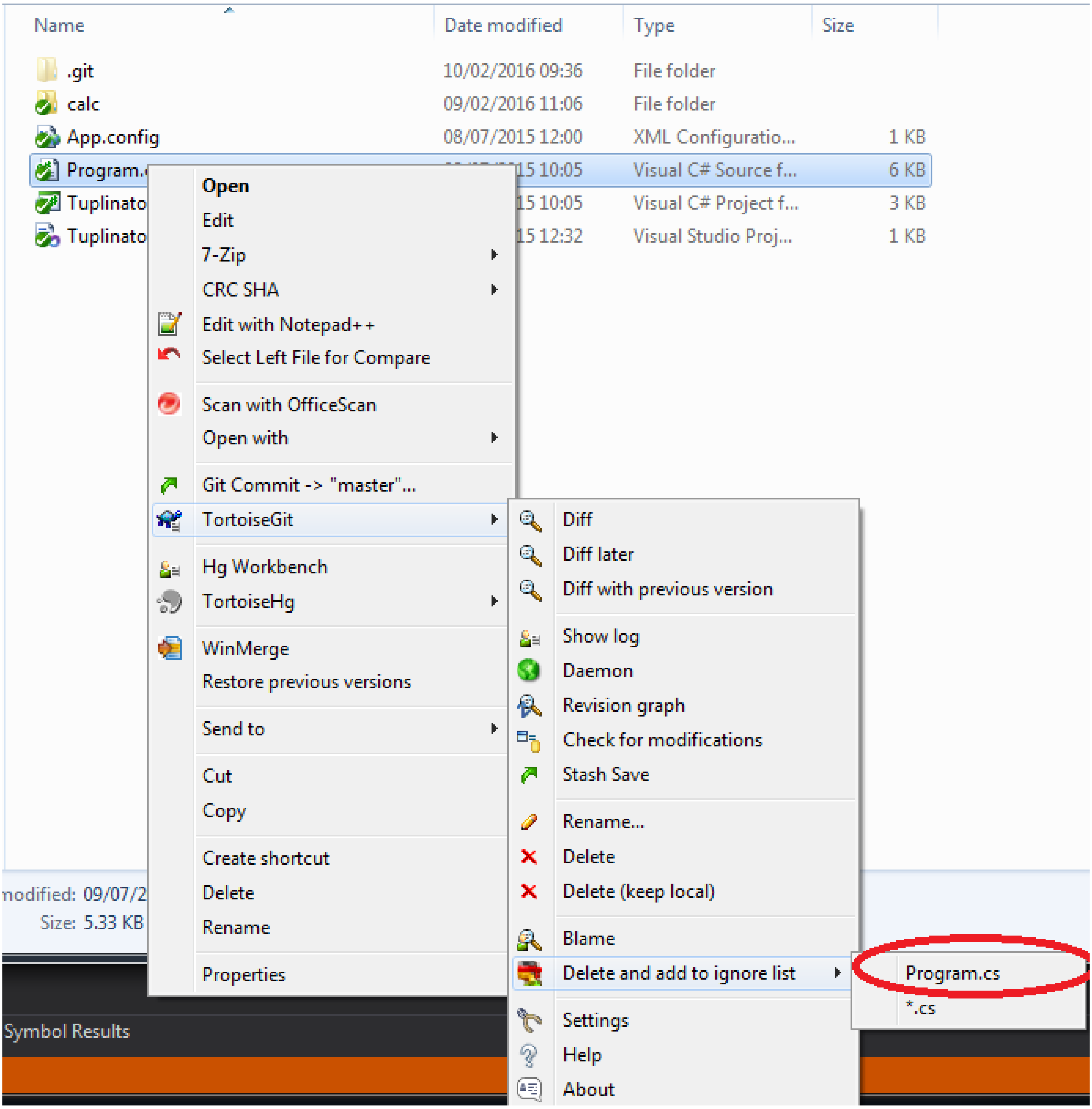
`git checkout -b [branch]`。 。

Reflog - git https://riptutorial.com/zh-TW/git/topic/5149/reflog----git

21: TortoiseGit

Examples

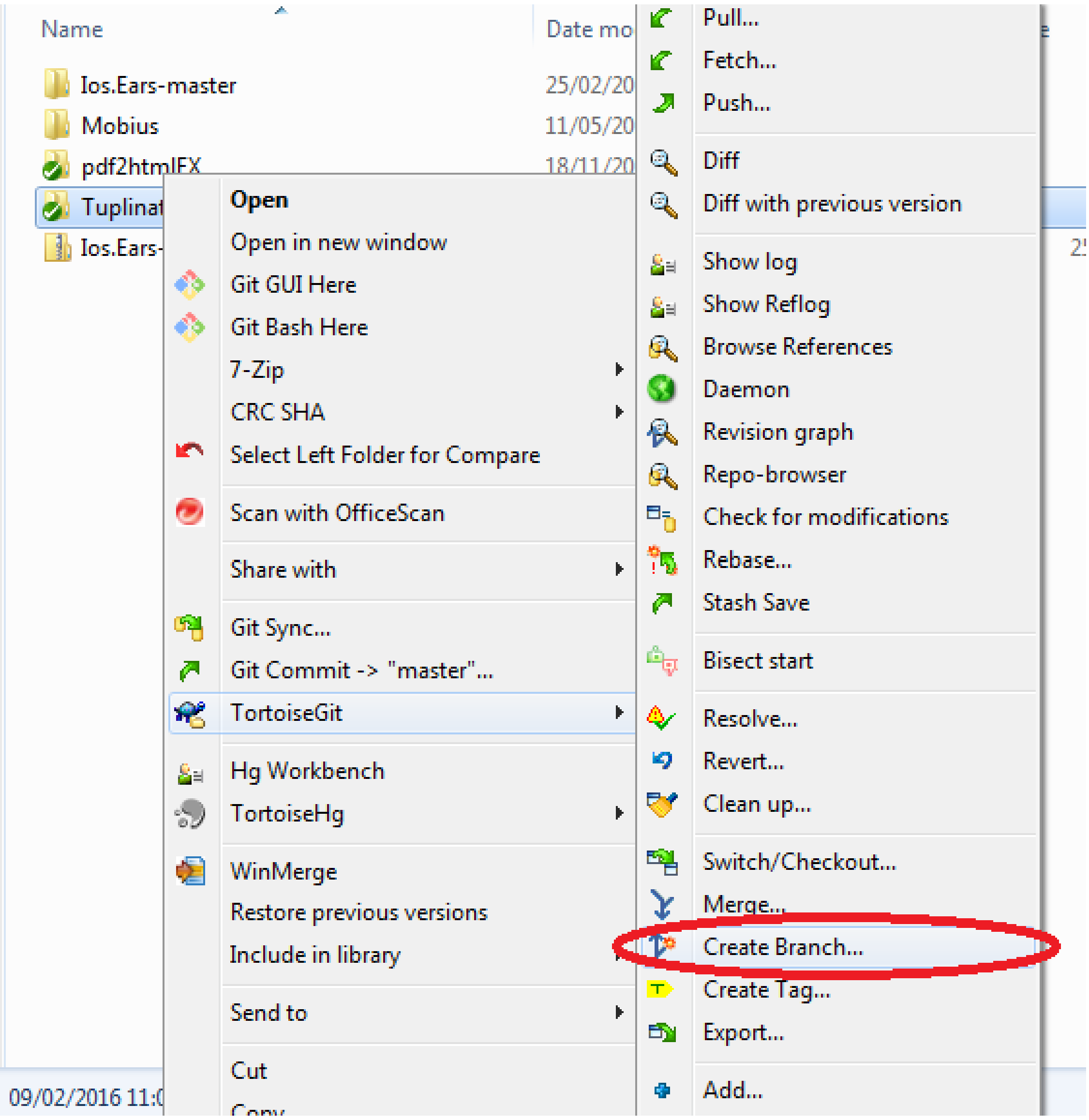
TortioseGit□□□□ - > `TortoiseGit` - > `Delete and add to ignore list` - >`Ok`。



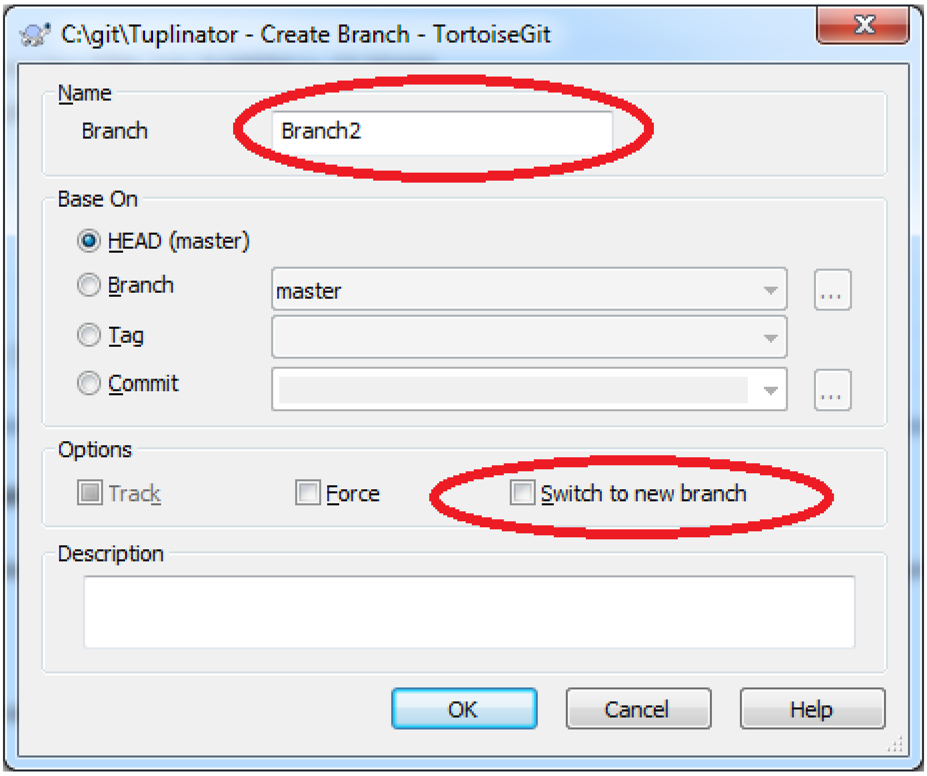
Name	Date modified	Type	Size
.git	10/02/2016 09:36	File folder	
calc	09/02/2016 11:06	File folder	
App.config	08/07/2015 12:00	XML Configuratio...	1 KB
Program.c	…15 10:05	Visual C# Source f...	6 KB
Tuplinato	…15 10:05	Visual C# Project f...	3 KB
Tuplinato	…15 12:32	Visual Studio Proj...	1 KB

nodified: 09/07/2
Size: 5.33 KB

UI□□□□ `Tortoise Git` - > `Create Branch...`



Give branch a name - > Give branch a name - >Switch to new branch。 - >OK。



C:\git\Tuplinator - Create Branch - TortoiseGit

Name

Branch: Branch2

Base On

● HEAD (master)

○ Branch: master

○ Tag

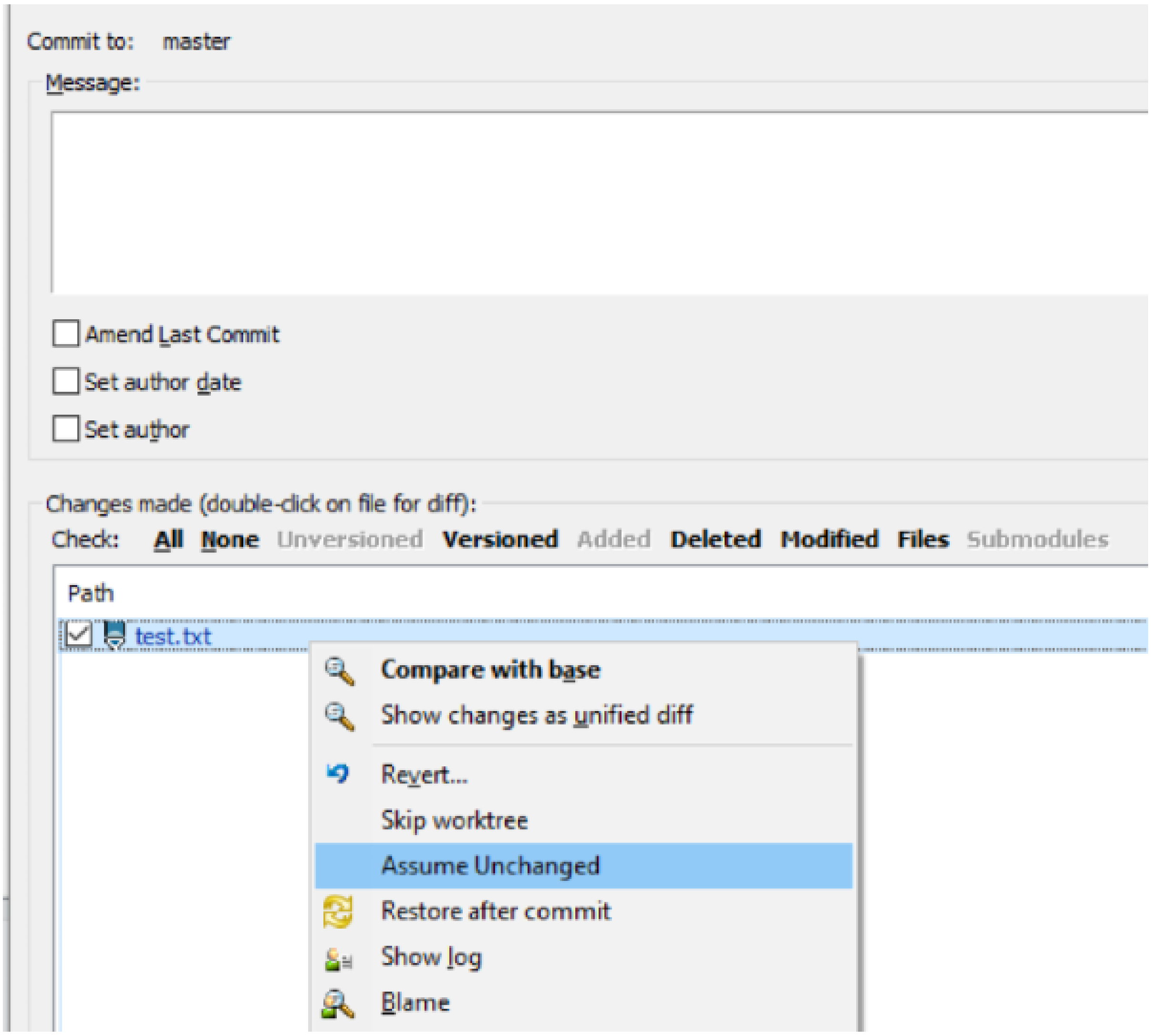
○ Commit

Options

☐ Track ☐ Force ☐ Switch to new branch

Description

“”



Commit to: master

Message:

☐ Amend Last Commit

☐ Set author date

☐ Set author

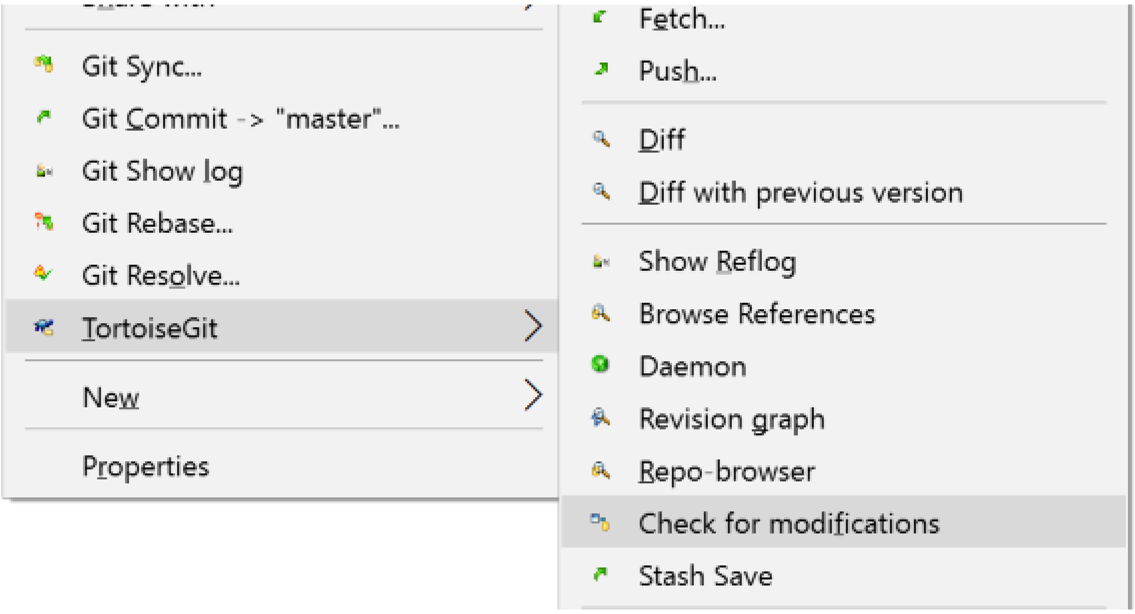
Changes made (double-click on file for diff):

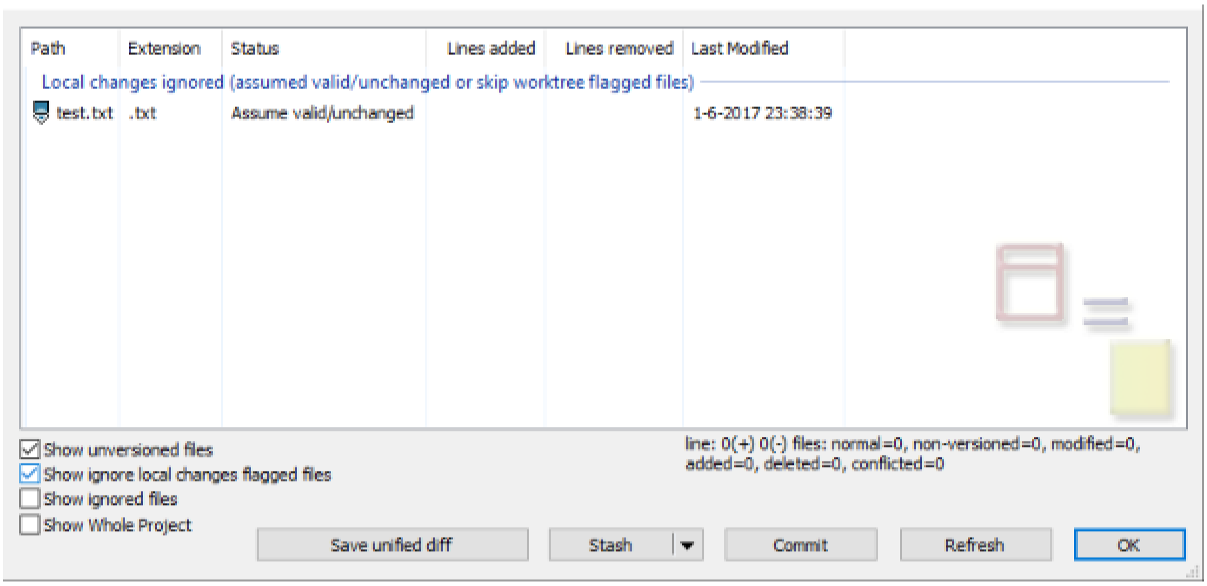
Check: **All** **None** Unversioned **Versioned** Added **Deleted** **Modified** **Files** Submodules

Path

☑ test.txt

“ ”

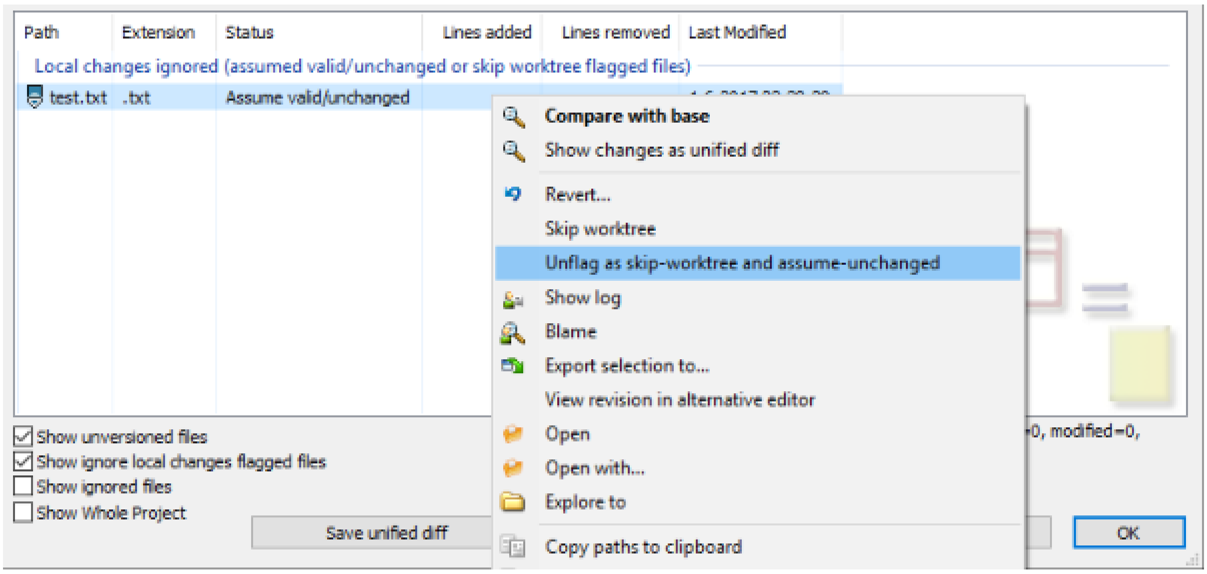




Path	Extension	Status	Lines added	Lines removed	Last Modified
Local changes ignored (assumed valid/unchanged or skip worktree flagged files)					
test.txt	.txt	Assume valid/unchanged			1-6-2017 23:38:39

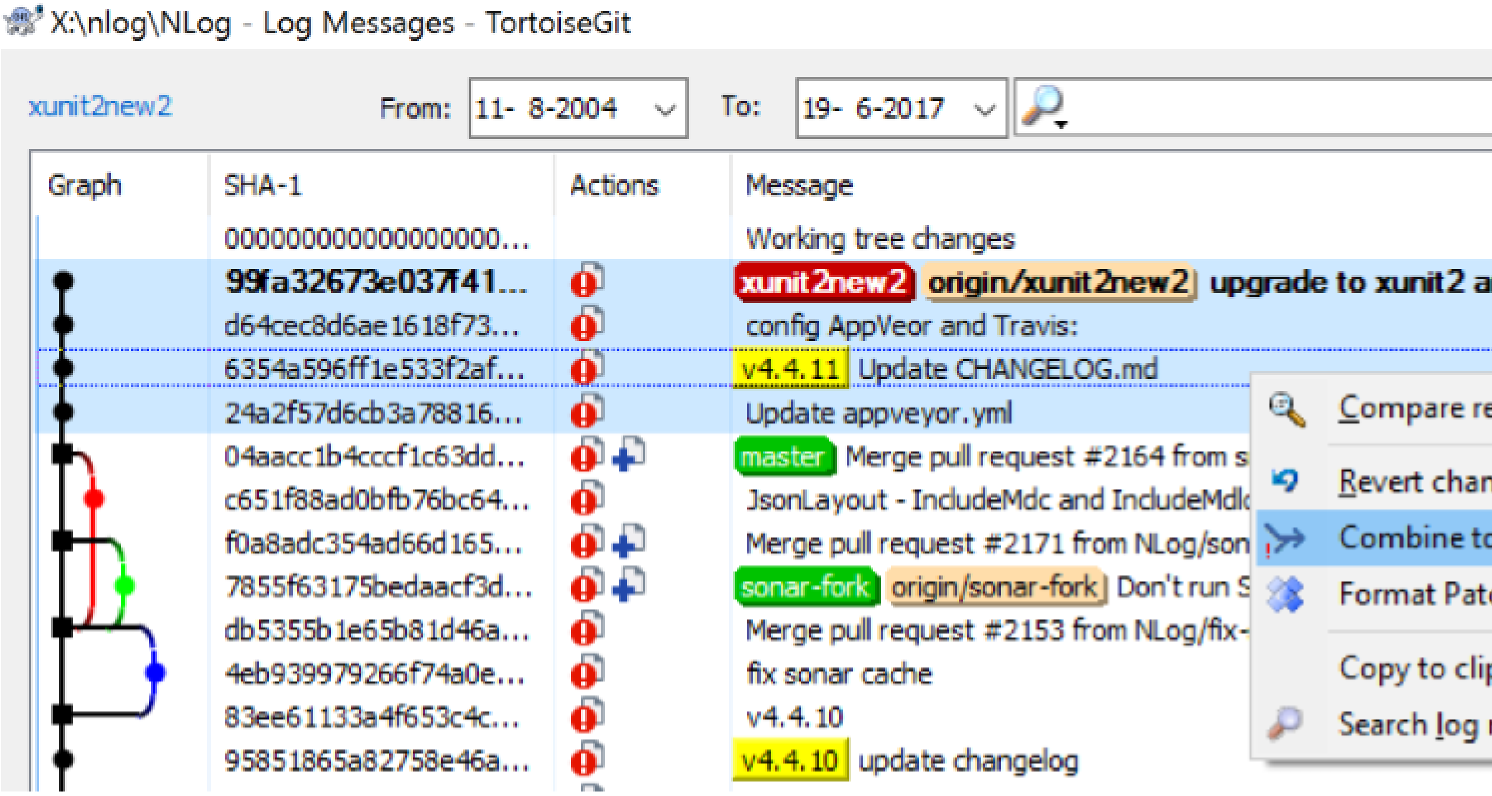
line: 0(+) 0(-) files: normal=0, non-versioned=0, modified=0, added=0, deleted=0, conflicted=0

☑ Show unversioned files
☑ Show ignore local changes flagged files
☐ Show ignored files
☐ Show Whole Project



Path	Extension	Status	Lines added	Lines removed	Last Modified
Local changes ignored (assumed valid/unchanged or skip worktree flagged files)					
test.txt	.txt	Assume valid/unchanged			

☑ Show unversioned files
☑ Show ignore local changes flagged files
☐ Show ignored files
☐ Show Whole Project

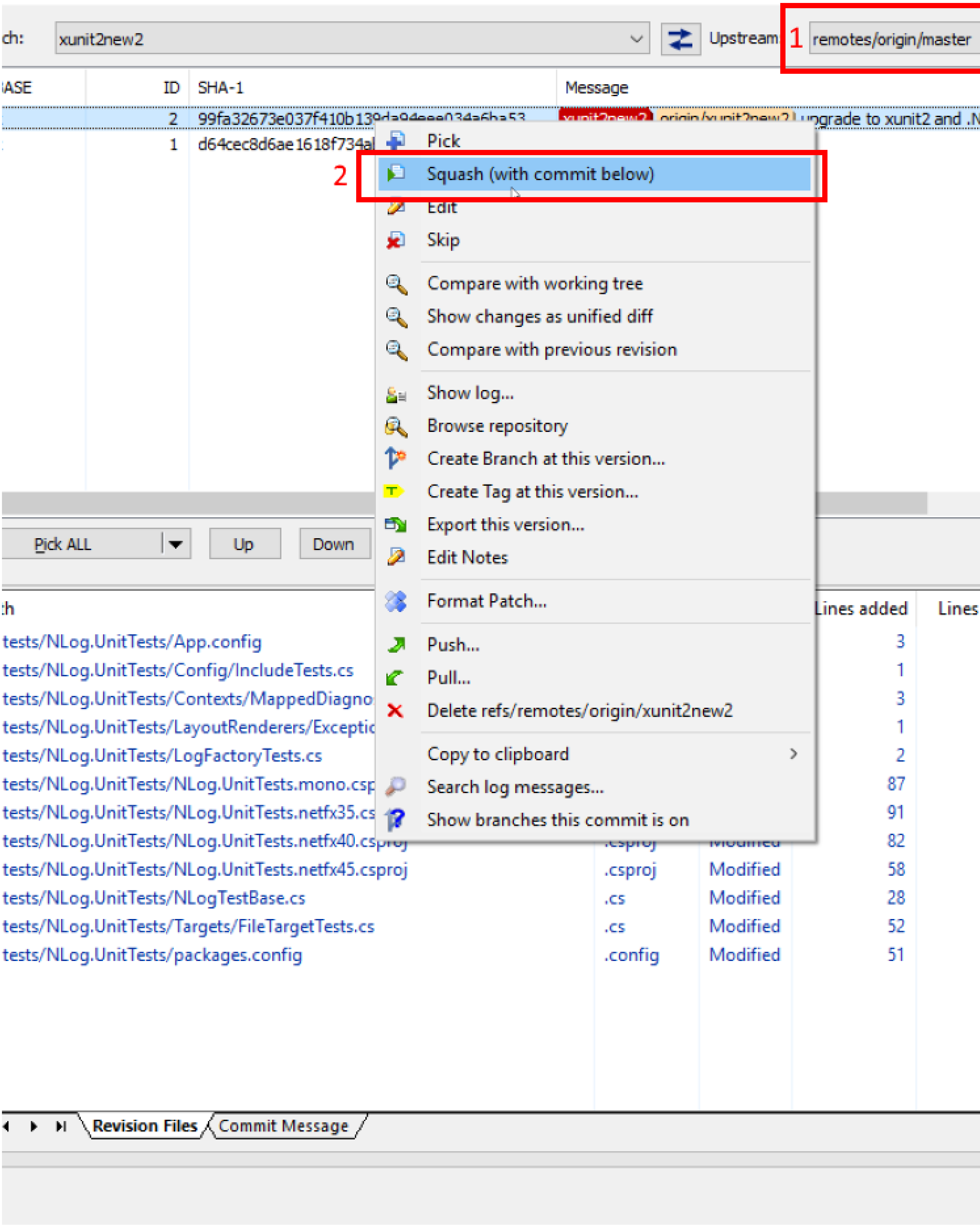


X:\nlog\NLog - Log Messages - TortoiseGit

xunit2new2 From: 11- 8-2004 To: 19- 6-2017

Graph	SHA-1	Actions	Message
	00000000000000000...		Working tree changes
●	**99fa32673e037f41...**		**xunit2new2** **origin/xunit2new2** **upgrade to xunit2 a**
●	d64cec8d6ae1618f73...		config AppVeor and Travis:
●	6354a596ff1e533f2af...		v4.4.11 Update CHANGELOG.md
●	24a2f57d6cb3a78816...		Update appveyor.yml
■	04aacc1b4cccf1c63dd...		master Merge pull request #2164 from s
●	c651f88ad0bfb76bc64...		JsonLayout - IncludeMdc and IncludeMdl
■	f0a8adc354ad66d165...		Merge pull request #2171 from NLog/son
●	7855f63175bedaacf3d...		sonar-fork origin/sonar-fork Don't run S
■	db5355b1e65b81d46a...		Merge pull request #2153 from NLog/fix-
●	4eb939979266f74a0e...		fix sonar cache
■	83ee61133a4f653c4c...		v4.4.10
●	95851865a82758e46a...		v4.4.10 update changelog

rebase


1
2
Pick
Squash (with commit below)
Edit
Skip
Compare with working tree
Show changes as unified diff
Compare with previous revision
Show log...
Browse repository
Create Branch at this version...
Create Tag at this version...
Export this version...
Edit Notes
Format Patch...
Push...
Pull...
Delete refs/remotes/origin/xunit2new2
Copy to clipboard
Search log messages...
Show branches this commit is on
Pick ALL
Up
Down
Revision Files
Commit Message

ch: xunit2new2

Upstream: remotes/origin/master

ASE	ID	SHA-1	Message
	2	99fa32673e037f410b139da94eee034a6ba53	xunit2new2 origin/xunit2new2 upgrade to xunit2 and .N
	1	d64cec8d6ae1618f734a	

th			Lines added	Lines
tests/NLog.UnitTests/App.config			3	
tests/NLog.UnitTests/Config/IncludeTests.cs			1	
tests/NLog.UnitTests/Contexts/MappedDiagno			3	
tests/NLog.UnitTests/LayoutRenderers/Exceptio			1	
tests/NLog.UnitTests/LogFactoryTests.cs			2	
tests/NLog.UnitTests/NLog.UnitTests.mono.csp			87	
tests/NLog.UnitTests/NLog.UnitTests.netfx35.cs			91	
tests/NLog.UnitTests/NLog.UnitTests.netfx40.csproj	.csproj	Modified	82	
tests/NLog.UnitTests/NLog.UnitTests.netfx45.csproj	.csproj	Modified	58	
tests/NLog.UnitTests/NLogTestBase.cs	.cs	Modified	28	
tests/NLog.UnitTests/Targets/FileTargetTests.cs	.cs	Modified	52	
tests/NLog.UnitTests/packages.config	.config	Modified	51	

TortoiseGit https://riptutorial.com/zh-TW/git/topic/5150/tortoisegit

22: Worktrees

- git worktree add [-f] [--detach] [--checkout] [-b <new-branch>] <path> [<branch>]
- git worktree prune [-n] [-v] [--expire <expire>]
- git worktree list [--porcelain]

-f --force	`<branch>`。 。
-b `<new-branch>` -B `<new-branch>`	`<new-branch><branch> <new-branch>`。 `<branch> HEAD` 。 `-b`。 `-B<new-branch><branch>` 。
-	add`HEAD` 。
- [-	`<branch>` `--no-checkout`。
-n --dry-run	;。
-	。 Git。
-v --verbose	。
--expire `<time>`	`<time>`。

https //git-scm.com/docs/git-worktree 。

Examples

。 `git stash`。 。

```
$ git worktree add -b emergency-fix ../temp master
$ pushd ../temp
# ... work work work ...
$ git commit -a -m 'emergency fix for boss'
$ popd
$ rm -rf ../temp
$ git worktree prune
```

。 `git mergegit format-patch`。

2.11.0。 https://git-scm.com/docs/git-worktree#_bugs 。

`.git`。

repo`/home/user/project-main /home/user/project-1 /home/user/project-2` 。

git。

1. worktree `.git`。 `/home/user/project-1/.git`

   ```
   gitdir: /home/user/project-main/.git/worktrees/project-2
   ```

2. worktree `.git`

   ```
   $ mv /home/user/project-main/.git/worktrees/project-1 /home/user/project-
   main/.git/worktrees/project-2
   ```

3. `/home/user/project-main/.git/worktrees/project-2/gitdir`。

   ```
   /home/user/project-2/.git
   ```

4.
   ```
   $ mv /home/user/project-1 /home/user/project-2
   ```

```
$ git worktree list
/home/user/project-main  23f78ad [master]
/home/user/project-2     78ac3f3 [branch-name]
```

`git worktree prune`。

Worktrees https://riptutorial.com/zh-TW/git/topic/3801/worktrees

23: .gitattributes

Examples

`.gitattributes`

```
* -text
```

`core.autocrlf = false` 。

`.gitattributes`

```
* text=auto
```

GitLF。

`core.autocrlf`

- Linux / macOS`input`
- Windows`true`

Git。 `.gitattributes`

```
*.png binary
```

`binary-diff -merge -text` 。

.gitattribute

`.gitattributes.gitattributes`

- https://gitattributes.io/
- https://github.com/alexkaratarakis/gitattributes

.gitattributes https://riptutorial.com/zh-TW/git/topic/1269/-gitattributes

24: filter-branch

Examples

◦ `$GIT_AUTHOR_NAME`◦

`filter.sh`

```
if [ "$GIT_AUTHOR_NAME" = "Author to Change From" ]
then
    export GIT_AUTHOR_NAME="Author to Change To"
    export GIT_AUTHOR_EMAIL="email.to.change.to@example.com"
fi
```

`filter-branch`

```
chmod +x ./filter.sh
git filter-branch --env-filter ./filter.sh
```

git committercommit author

`commit1..commit2` git commit authorgit committer

```
git filter-branch -f --commit-filter \
   'export GIT_COMMITTER_NAME=\"$GIT_AUTHOR_NAME\";
    export GIT_COMMITTER_EMAIL=\"$GIT_AUTHOR_EMAIL\";
    export GIT_COMMITTER_DATE=\"$GIT_AUTHOR_DATE\";
    git commit-tree $@' \
    -- commit1..commit2
```

filter-branch https://riptutorial.com/zh-TW/git/topic/2825/filter-branch

25: Gitk

Examples

```
gitk path/to/myfile
```

`d9e1db95651067` 。 `d9e1db9` `5651067`。

```
gitk --ancestry-path d9e1db9 5651067
```

`v2.3` 。

```
gitk v2.3..
```

Gitk https://riptutorial.com/zh-TW/git/topic/3637/gitk

26:

- `git remote [-v | --verbose]`
- `git remote add [-t <branch>] [-m <master>] [-f] [--[no-]tags] [--mirror=<fetch|push>] <name> <url>`
- `git remote rename <old> <new>`
- `git remote remove <name>`
- `git remote set-head <name> (-a | --auto | -d | --delete | <branch>)`
- `git remote set-branches [--add] <name> <branch>…`
- `git remote get-url [--push] [--all] <name>`
- `git remote set-url [--push] <name> <newurl> [<oldurl>]`
- `git remote set-url --add [--push] <name> <newurl>`
- `git remote set-url --delete [--push] <name> <url>`
- `git remote [-v | --verbose] show [-n] <name>…`
- `git remote prune [-n | --dry-run] <name>…`
- `git remote [-v | --verbose] update [-p | --prune] [(<group> | <remote>)…]`

Examples

```
git remote add upstream git-repository-url
```

`git-repository-url`gitgit`git-repository-url``upstream`

“”

```
git fetch remote-name
git merge remote-name/branch-name
```

`pull``fetch``merge`。

```
git pull
```

`pull` with `--rebase` flag`fetch``rebase``merge`。

```
git pull --rebase remote-name branch-name
```

LS-

`git ls-remote`/ 。

refs / headsrefs / tags。

`refs/tags/v0.1.6` `refs/tags/v0.1.6^{}` `^{}`

git 2。 820163

```
git ls-remote --ref
```

“ `url.<base>.insteadOf` ”URL。

`git remote --get-url <aremotename>`https://server.com/user/repo `git config url.ssh://git@server.com:.insteadOf https://server.com/`

```
git ls-remote --get-url <aremotename>
ssh://git@server.com:user/repo
```

Git

```
git push [remote-name] --delete [branch-name]
```

```
git push [remote-name] :[branch-name]
```

。

```
git fetch [remote-name] --prune
```

```
git fetch --all --prune
```

`origin`

```
git remote show origin
```

URL

```
git config --get remote.origin.url
```

2.7+`config`。

```
git remote get-url origin
```

```
git remote
```

`fetchpush` URL

```
git remote --verbose
```

```
git remote -v
```

`git push <remote_name> <branch_name>`

`git push origin master`

```
git checkout -b AP-57
```

git checkout

```
git push --set-upstream origin AP-57
```

git push。

URL`set-url`

```
git remote set-url <remote_name> <remote_repository_url>
```

```
git remote set-url heroku https://git.heroku.com/fictional-remote-repository.git
```

GitURL

```
git remote -v
# origin https://github.com/username/repo.git (fetch)
# origin https://github.com/usernam/repo.git (push)
```

URL

```
git remote set-url origin https://github.com/username/repo2.git
# Change the 'origin' remote's URL
```

URL

```
git remote -v
# origin  https://github.com/username/repo2.git (fetch)
# origin  https://github.com/username/repo2.git (push)
```

`git remote rename`

`git remote rename`

- **origin**
- **destination**

```
git remote
# origin
```

URL

```
git remote -v
# origin https://github.com/username/repo.git (fetch)
# origin https://github.com/usernam/repo.git (push)
```

```
 git remote rename origin destination
 # Change remote name from 'origin' to 'destination'
```

```
git remote -v
# destination https://github.com/username/repo.git (fetch)
# destination https://github.com/usernam/repo.git (push)
```

======

1. 'remote。 [old name]''remote。 [new name]'

   。

2. []。

   。

URL

URL

```
git remote set-url remote-name url
```

URL

URL

`git remote get-url <name>`

`git remote get-url origin`

https://riptutorial.com/zh-TW/git/topic/243/

27:

- git clone [<options>] [-] <repo> [<dir>]
- git clone [--template = <template_directory>] [-l] [-s] [--no-hardlinks] [-q] [-n] [--bare] [--mirror] [-o <name>] [-b <name>] [-u <upload-pack>] [- reference <repository>] [--disociate] [--separate-git-dir <git dir>] [--depth <depth>] [- [no-] single-branch] [--recursive | --recurse-submodules] [- [no-] shallow-submodules] [--jobs <n>] [-] <repository> [<directory>]

Examples

◦

```
git clone [repo_url] --depth 1
```

◦

◦ ◦ 50

```
git clone [repo_url] --depth 50
```

1.8.3

```
git fetch --unshallow     # equivalent of git fetch --depth=2147483647
                          # fetches the rest of the repository
```

1.8.3

```
git fetch --depth=1000    # fetch the last 1000 commits
```

```
git clone <url>
```

◦

```
git clone <url> [directory]
```

◦

URL`--branch <branch name>`

```
git clone --branch <branch name> <url> [directory]
```

`--branch`-`b` ◦ `<branch name>` ◦

```
git clone --branch <branch_name> --single-branch <url> [directory]
```

`--single-branch[directory]` 。 。

`--single-branch`

```
git config remote.origin.fetch "+refs/heads/*:refs/remotes/origin/*"
git fetch origin
```

1.6.5

```
git clone <url> --recursive
```

。 Git。

git。

```
git config --global http.proxy http://<proxy-server>:<port>/
```

https://riptutorial.com/zh-TW/git/topic/1405/

28:

Examples

git repository。

.git/。 git。 .git/。

```
.git/
    objects/
    refs/
```

git。 gitobjectobjectSHA-1。

git

`git cat-file -p 4bb6f98`

Object4

- blob
- tree
- commit
- tag

HEAD

HEADref 。 。

.git/HEAD。

HEADref

```
$cat .git/HEAD
ref: refs/heads/mainline
```

object

```
$ cat .git/HEAD
4bb6f98a223abc9345a0cef9200562333
```

“” - HEADref object 。

ref。 object。

```
"master" --> 1a410e...
```

`.git / refs / heads /。

```
$ cat .git/refs/heads/mainline
4bb6f98a223abc9345a0cef9200562333
```

branches。 gitbranch - ref。

objectsgit。 。 refobjects。 git。

commitgitobjectgit commitobject。

commit。 objectsobjectscommit。

commit

- tree
- commit
- //
-

```
$ git cat-file commit 5bac93
tree 04d1daef...
parent b7850ef5...
author Geddy Lee <glee@rush.com>
commiter Neil Peart <npeart@rush.com>

First commit!
```

---

tree。 commit*。

* 。 。 *commit* 。

---

parentcommit“”“”。 。 DAG。

tree。

tree

- 0blob
- 0tree

lsdirtree。

```
$ git cat-file -p 07b1a631
100644 blob b91bba1b   .gitignore
100644 blob cc0956f1   Makefile
040000 tree 92e1ca7e   src
...
```

committreecommit tree

```
$ git cat-file commit 4bb6f93a
tree 07b1a631
parent ...
author ...
commiter ...

$ git cat-file -p 07b1a631
100644 blob b91bba1b   .gitignore
100644 blob cc0956f1   Makefile
040000 tree 92e1ca7e   src
...
```

Blob

blob。 。 PNG。

blob。

```
$ git cat-file -p d429810
package com.example.project

class Foo {
 ...
}
...
```

tree blobs 。

```
$ git cat-file -p 07b1a631
100644 blob b91bba1b   .gitignore
100644 blob cc0956f1   Makefile
040000 tree 92e1ca7e   src
100644 blob cae391ff   Readme.txt

$ git cat-file -p cae391ff
Welcome to my project! This is the readmefile
...
```

git commit

1. blobstrees - .git/objects
2. 1treecommit - .git/objects
3. .git/HEADHEAD refcommit

git 。

HEAD

hashrefgit checkout git。

1. committree
2. HEADref

`git reset --hard`refshash / ref。

`MyBranch``b8dc53`

```
$ git checkout MyBranch      # moves HEAD to MyBranch
$ git reset --hard b8dc53    # makes MyBranch point to b8dc53
```

`git checkout -b <refname>`commit。

```
$ cat .git/head
1f324a

$ git checkout -b TestBranch

$ cat .git/refs/heads/TestBranch
1f324a
```

https://riptutorial.com/zh-TW/git/topic/2637/

29:

“”。 git。

cherry-pickdiff。 。

。 。

。 。 。 CSS。 。 。 。

Examples

```
git add <filename>
```

```
git add -A
```

2.0

```
git add .
```

2.x `git add .`。 1.x 。

`git add -A` `git add --all` git。

```
git rm filename
```

git`--cached`

```
git rm --cached filename
```

```
git reset <filePath>
```

`git add -i --interactive` 。 。 Git`git gui` ;。

12rebase。

```
$ git add -i
           staged     unstaged path
  1:    unchanged        +4/-4 index.js
  2:        +1/-0      nothing package.json

*** Commands ***
  1: status       2: update       3: revert       4: add untracked
  5: patch        6: diff         7: quit         8: help
What now>
```

1. `index.js`44。 “”。 `+4/-4`“”。
2. `package.json`。 “”。

。 1-8 `s u r a p d q h`。

`status`。

`update`。

`revert`HEAD。

`add untracked`。

`patchstatus`。

`diff`。

`quit`。

`help`。

hunk

patch“”

```
git add -p
```

```
git add --patch
```

。

```
Stage this hunk [y,n,q,a,d,/,s,e,?]?
```

- `y`
- `n`
- `Q`;
- ▯▯
- `d`
- `g`
- `/`
- `j`
- `J`
- `k`
- `K`
- ▯
- `e`
- ▯

。

git add --interactivep。

```
git diff --cached
```

https://riptutorial.com/zh-TW/git/topic/244/

30:

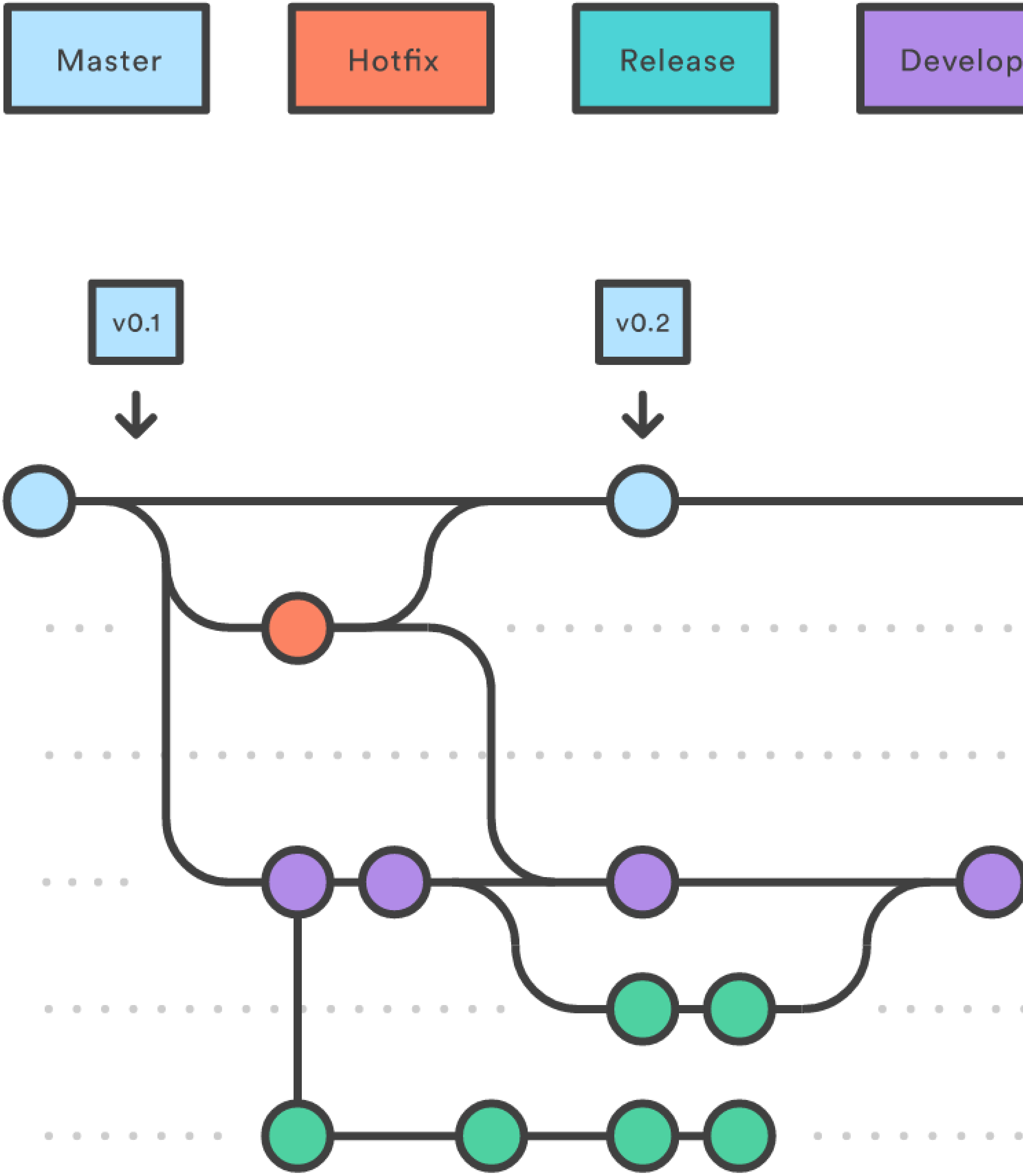
Git。 。

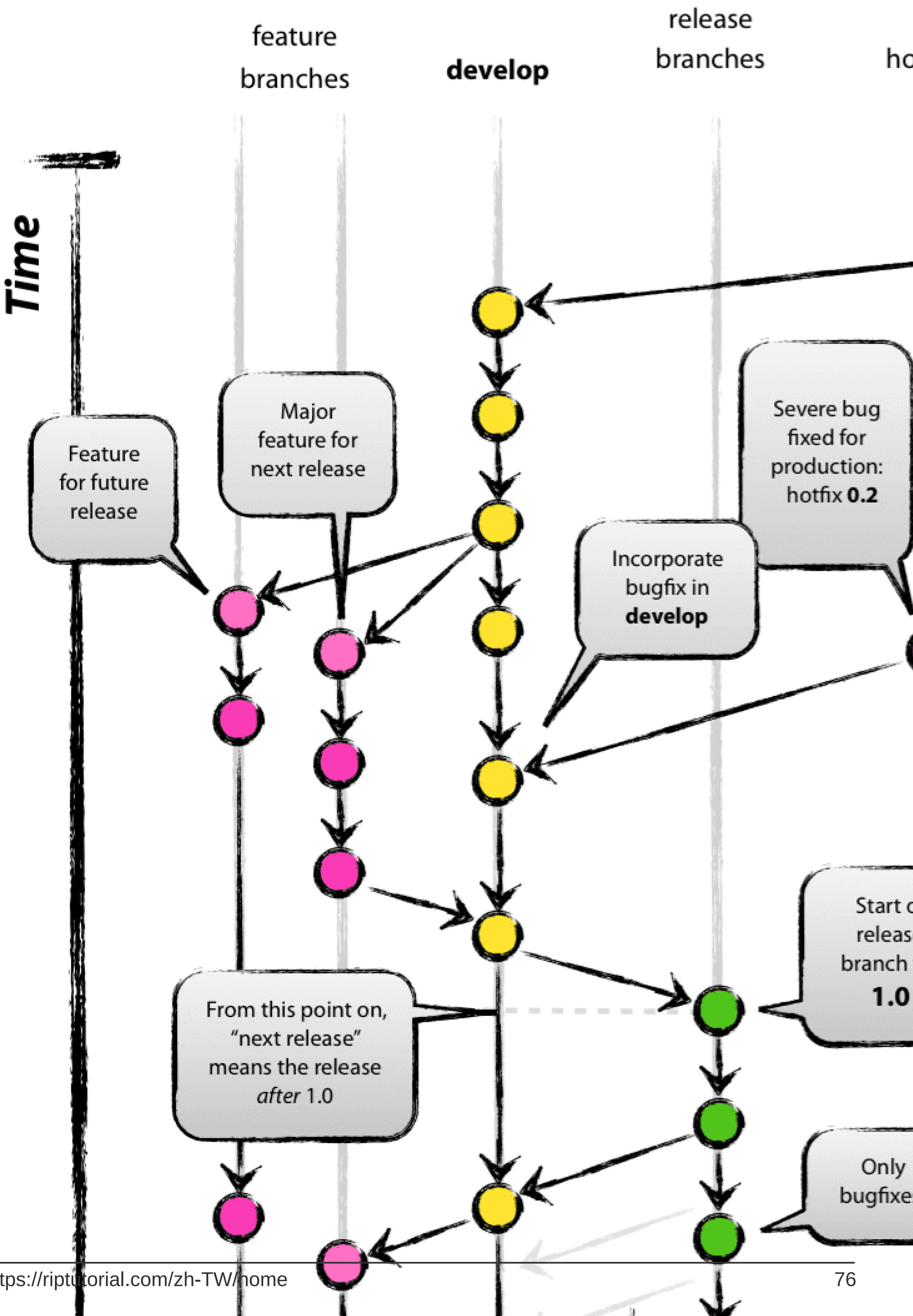
Examples

Gitflow

Vincent DriessenGitflowgit。 。

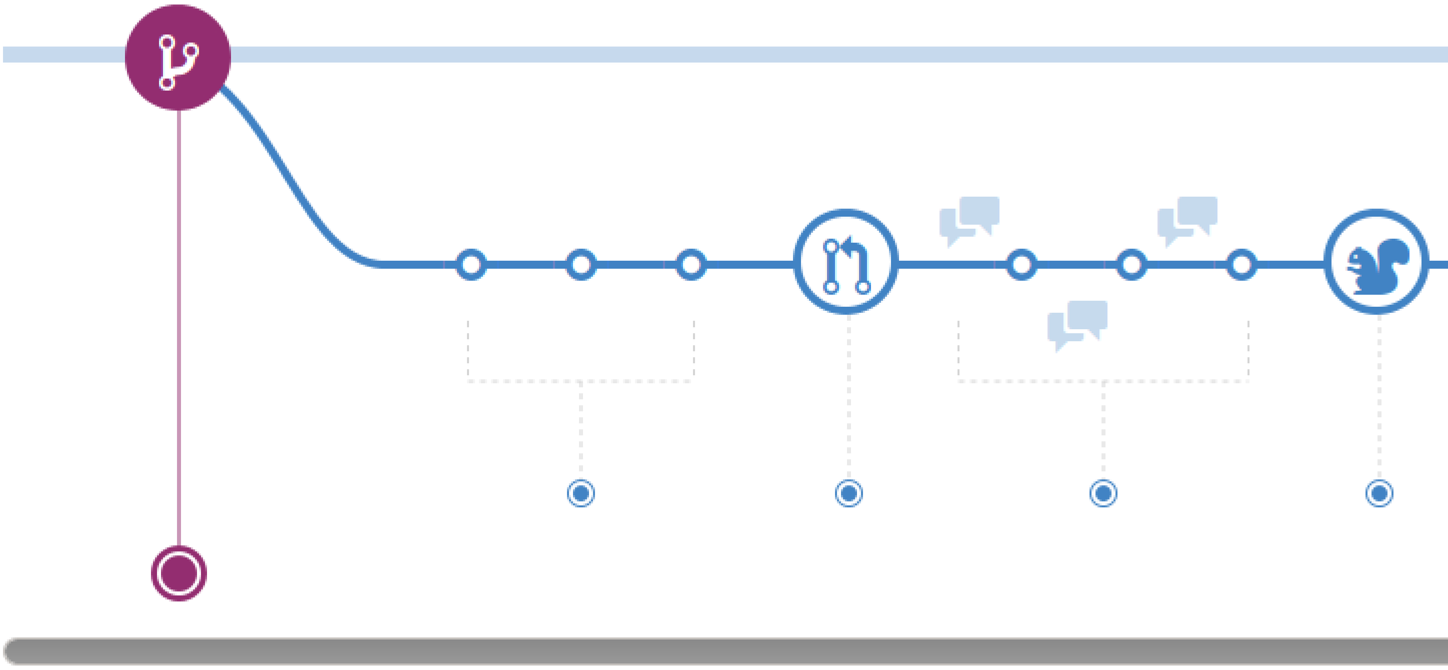
- `master`。 。
- `develop` 。 。 /`release`。
- `hotfix`。 `hotfix` `master` `master` `develop` 。
- `release` `develop` `master`。 `master` `develop` 。 `master` 。
- `feature`。 `develop`。 `feature`。


Master
Hotfix
Release
Develop
v0.1
v0.2


feature branches
develop
release branches
ho
Time
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Start o releas branch 1.0
From this point on, "next release" means the release after 1.0
Only bugfixe

- masternew-oauth2-scopes
-
-
- master
- “”

Scott Chacon 。



GitHub Flow

https://riptutorial.com/zh-TW/git/topic/1276/

31:

- `git branch [--set-upstream | --track | --no-track] [-l] [-f] <branchname> [<start-point>]`
- `git branch (--set-upstream-to=<upstream> | -u <upstream>) [<branchname>]`
- `git branch --unset-upstream [<branchname>]`
- `git branch (-m | -M) [<oldbranch>] <newbranch>`
- `git branch (-d | -D) [-r] <branchname>…`
- `git branch --edit-description [<branchname>]`
- `git branch [--color[=<when>] | --no-color] [-r | -a] [--list] [-v [--abbrev=<length> | --no-abbrev]] [--column[=<options>] | --no-column] [(--merged | --no-merged | --contains) [<commit>]] [--sort=<key>] [--points-at <object>] [<pattern>…]`

-d -	◦ `--track--set-upstreamHEAD`
-D	`--delete --force`
-m -	/reflog
-M	`--move --force`
-r -	-d
-a - all	
--list	◦ `git branch <pattern>git branch --list <pattern>`
--set	`--force --track`◦ --track

git ◦ `HEAD`◦

git repo`git branch*git commitHEAD` HEAD◦

`HEAD` ◦

Examples

Git◦ `git branch`◦ Git◦

	`git branch`
	`git branch -v`
	`git branch -agit branch --all`
	`git branch -av`
	`git branch -r`
	`git branch -rv`

	git branch --merged
	git branch --no-merged
	git branch --contains [<commit>]

- v-v$ git branch -avv$ git branch -vv。
-

```
git branch <name>
```

。

```
git checkout <name>
```

```
git checkout -b <name>
```

HEAD

```
git branch <name> [<start-point>]
git checkout -b <name> [<start-point>]
```

`<start-point>`git SHAHEAD

```
git checkout -b <name> some_other_branch
git checkout -b <name> af295
git checkout -b <name> HEAD~5
git checkout -b <name> v1.0.5
```

`<remote_name>`origin

```
git branch <name> <remote_name>/<branch_name>
git checkout -b <name> <remote_name>/<branch_name>
```

```
git checkout -b <branch_name>
```

```
git checkout -b <branch_name> <remote_name>/<branch_name>
```

。“”

```
git branch <new_name>
git reset --hard HEAD~2 # Go back 2 commits, you will lose uncommitted work.
git checkout <new_name>
```

```
 Initial state       After git branch <new_name>    After git reset --hard HEAD~2
                             newBranch                        newBranch
                                 ↓                                ↓
A-B-C-D-E (HEAD)         A-B-C-D-E (HEAD)                 A-B-C-D-E (HEAD)
        ↑                        ↑                             ↑
```

```
      master                        master                           master
```

```
$ git branch -d dev
```

dev。 devgit branch -d

```
$ git branch -d dev
error: The branch 'dev' is not fully merged.
If you are sure you want to delete it, run 'git branch -D dev'.
```

-D

```
$ git branch -D dev
```

feature origin/feature

- git checkout --track -b feature origin/feature
- git checkout -t origin/feature
- git checkout feature - featurefeaturefeature 。

-

- git branch --set-upstream-to=<remote>/<branch> <branch>
- git branch -u <remote>/<branch> <branch>

- <remote> origin develop
- <branch>。

- git branch -vv

```
git branch -m new_branch_name
```

```
git branch -m branch_you_want_to_rename new_branch_name
```

。

file.example path/to/ 。

```
git checkout some-branch path/to/file
```

*some-branch*git*tree-ish* gitrevisions

--。 --。

```
git checkout some-branch -- some-file
```

some-file。

`origin`Git 1.5.0

```
git push origin :<branchName>
```

Git 1.7.0

```
git push origin --delete <branchName>
```

```
git branch --delete --remotes <remote>/<branch>
git branch -dr <remote>/<branch> # Shorter

git fetch <remote> --prune # Delete multiple obsolete tracking branches
git fetch <remote> -p      # Shorter
```

。

```
git branch -d <branchName>
```

```
git branch -D <branchName>
```

```
git checkout --orphan new-orphan-branch
```

。

。

`git push`

- `origin`
- `master`

```
git push  <REMOTENAME> <BRANCHNAME>
```

`git push origin master`。

`-u` `--set-upstream` set `--set-upstream`。

```
git push -u <REMOTENAME> <BRANCHNAME>
```

`git`。 `new-feature` `new-feature` 。 `:`

```
git push <REMOTENAME> <LOCALBRANCHNAME>:<REMOTEBRANCHNAME>
```

HEAD

。 `aabbcc`

```
git reset --hard aabbcc
```

。。reflog。。

。

```
git checkout -
```

```
git branch --contains <commit>
```

```
git branch -a --contains <commit>
```

https://riptutorial.com/zh-TW/git/topic/415/

32:

Examples

Git

- `~/.gitconfig`

```
[alias]
    ci = commit
    st = status
    co = checkout
```

-

```
git config --global alias.ci "commit"
git config --global alias.st "status"
git config --global alias.co "checkout"
```

-

- `git ci` `git commit`
- `git st` `git status`
- `git co` `git checkout` 。

git。

```
git ci -m "Commit message..."
git co -b feature-42
```

/

`--get-regexp` git

```
$ git config --get-regexp '^alias\.'
```

`[alias]` `.gitconfig`

```
aliases = !git config --list | grep ^alias\\. | cut -c 7- | grep -Ei --color \"$1\" "#"
```

- `git aliases` -
- `git aliases commit` - “commit”

Gitgit`sh` shell`!` 。

`~/.gitconfig`

```
[alias]
    temp = !git add -A && git commit -m "Temp"
```

shellshell

```
[alias]
    ignore = "!f() { echo $1 >> .gitignore; }; f"
```

`f`。 `git ignore .tmp/` `.tmp/` `.gitignore`。

Git`$1` `$2`。 Git。

`!`git checkout。 `cd`。

```
[alias]
    ignore = "! echo $1 >> .gitignore"
```

-

```
unwatch = update-index --assume-unchanged
```

```
watch = update-index --no-assume-unchanged
```

```
unwatched = "!git ls-files -v | grep '^[[:lower:]]'"
```

-

```
watchall = "!git unwatched | xargs -L 1 -I % sh -c 'git watch `echo % | cut -c 2-`'"
```

```
git unwatch my_file.txt
git watch my_file.txt
git unwatched
git watchall
```

```
[alias]
  logp=log --pretty=format:'%h %ad | %s%d [%an]' --graph --date=short

  lg =  log --graph --date-order --first-parent \
     --pretty=format:'%C(auto)%h%Creset %C(auto)%d%Creset %s %C(green)(%ad) %C(bold
cyan)<%an>%Creset'
  lgb = log --graph --date-order --branches --first-parent \
     --pretty=format:'%C(auto)%h%Creset %C(auto)%d%Creset %s %C(green)(%ad) %C(bold
cyan)<%an>%Creset'
 lga = log --graph --date-order --all \
   --pretty=format:'%C(auto)%h%Creset %C(auto)%d%Creset %s %C(green)(%ad) %C(bold
cyan)<%an>%Creset'
```

`--pretty` `git help log`

--graph -

--date-order -

--first-parent - 。

--branches -

--all -

h -

ad -

an -

an -

C - []

Creset -

d - - decrate

s -

ad - --date

an - cn

。 `git pull` 。

```
[alias]
  up = pull --rebase
```

。

```
git up
```

.gitignore

```
[ alias ]

    ignored = ! git ls-files --others --ignored --exclude-standard --directory \
            && git ls-files --others -i --exclude-standard
```

grep

```
$ git ignored | grep '/$'
.yardoc/
doc/
```

```
~$ git ignored | wc -l
199811                  # oops, my home directory is getting crowded
```

Unstage

`git reset` commit `reset`。 git`reset` `reset`。

`git config --global alias.unstage "reset --"`

`git unstage` `git unstage`。

https://riptutorial.com/zh-TW/git/topic/337/

33:

- git merge **another_branch** [options]
- git merge **--abort**

`-m`	
`-v`	
`--abort`	
`--ff-only`	
`--no-ff`	
`--no-commit`	
`--stat`	diffstat
`-n` / `--no-stat`	diffstat
`--squash`	

Examples

```
git merge incomingBranch
```

`incomingBranch`◦ `master` `incomingBranch``master` ◦

◦ `Automatic merge failed; fix conflicts and then commit the result.`

```
git merge --abort
```

Git

```
~/Stack Overflow(branch:master) » git merge another_branch
Auto-merging file_a
Merge made by the 'recursive' strategy.
 file_a | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

◦ `--abort`

```
git merge --abort
```

`--ours` - `--theirs``git checkout` ◦

```
$ git checkout --ours   -- file1.txt # Use our version of file1, delete all their changes
$ git checkout --theirs -- file2.txt # Use their version of file2, delete all our changes
```

。 `--no-ff`。

`git merge <branch_name> --no-ff -m "<commit message>"`

master。 `mastermaster`。 PRmaster。

```
 for branch in $(git branch -r) ; do
   [ "${branch}" != "origin/master" ] && [ $(git diff master...${branch} | wc -l) -eq 0 ] &&
echo -e `git show --pretty=format:"%ci %cr" $branch | head -n 1`\\t$branch
 done | sort -r
```

https://riptutorial.com/zh-TW/git/topic/291/

34:

Examples

```
git update-ref [-m <reason>] (-d <ref> [<oldvalue>] | [--no-deref] [--create-reflog] <ref>
<newvalue> [<oldvalue>] | --stdin [-z])
```

1. 。

   ```
   git update-ref HEAD <newvalue>
   ```

2. `refoldvaluenewvalueref`。

   ```
   git update-ref refs/head/master <newvalue> <oldvalue>
   ```

   `oldvalue`master`newvalue` 。

`<oldvalue>-d<ref> <oldvalue>` 。

`--create-reflog` update-refrefreflogreflog。

`-z`NULupdatecreatedeleteverify。

`<oldvalue> <ref><newvalue> <oldvalue>` 。 `<newvalue>`ref/`<oldvalue>`ref。

`<newvalue><ref>` 。 `<newvalue>`。

`<oldvalue><ref>` 。 `<oldvalue>`。

`<ref><oldvalue>`。 `<oldvalue>`ref。

https://riptutorial.com/zh-TW/git/topic/7579/

35:

- `git rebase [-i | --interactive] [options] [--exec <cmd>] [--onto <newbase>] [<upstream>] [<branch>]`
- `git rebase [-i | --interactive] [options] [--exec <cmd>] [--onto <newbase>] --root [<branch>]`
- `git rebase --continue | --skip | --abort | --edit-todo`

-	。
--abort	rebaseHEAD。 rebaseHEAD。 HEADrebase。
--keep	。
-	。
-m - merge	rebase。 rebase。 rebase。 。 。
--stat	rebase。 diffstatrebase.stat。
-x - exec `command`	rebase`command`

rebase。

。 。

Examples

。

`rebaserebase。`

```
git checkout topic
git rebase master  # rebase current branch onto master branch
```

```
      A---B---C topic
     /
D---E---F---G master
```

```
              A'--B'--C' topic
             /
D---E---F---G master
```

```
git rebase master topic   # rebase topic branch onto master branch
```

rebase。 。 `git pushgit pushgit push --force。`

Rebase

rebase“”“”

```
git checkout topic
git rebase   master    # rebase topic branch on top of master branch
```

HEAD“”

rebase`HEADmaster` ;cherry-picking`topictopictopic`。

```
=> local is master ("ours"),
=> remote is topic ("theirs")
```

/`local` `master` `remote` `topic`

```
+-----------------------------------------+
| LOCAL:master |    BASE   | REMOTE:topic |
+-----------------------------------------+
|             MERGED                      |
+-----------------------------------------+
```

```
c--c--x--x--x(*) <- current branch topic ('*'=HEAD)
    \
     \
      \--y--y--y <- other branch to merge
```

`topic`

```
c--c--x--x--x---------o(*)  MERGE, still on branch topic
    \       ^        /
     \     ours     /
      \            /
       --y--y--y--/
               ^
              theirs
```

---

rebaserebase

```
c--c--x--x--x(*) <- current branch topic ('*'=HEAD)
    \
     \
      \--y--y--y <- upstream branch
```

`git rebase upstream`**`git rebase upstream`**`HEAD`“”“”“”。

```
c--c--x--x--x <- former "current" branch, new "theirs"
    \
     \
      \--y--y--y(*) <- set HEAD to this commit, to replay x's on it
               ^       this will be the new "ours"
               |
            upstream
```

rebase“” `topic`“”

```
c--c..x..x..x <- old "theirs" commits, now "ghosts", available through "reflogs"
    \
     \
      \--y--y--y--x'--x'--x'(*) <- topic  once all x's are replayed,
               ^                      point branch topic to this commit
               |
        upstream branch
```

Rebase

`git rebase` 。 `git rebase`。

rebase

```
git rebase -i
```

`-i` 。 rebase/。

。

```
git rebase -i HEAD~3
```

。 。 rebase。 `git log`。

。 。

```
git rebase -i HEAD~3
```

`pick` `reword`。

rebase。 。 。

。 `pick``edit`。 Git。 。

。 。

。 `pick``edit`Enter。

git。 `git reset HEAD^`。 - *n*。

。 `git rebase -i HEAD~3` 3 。

`squashpick` 。 rebase;。

Rebase

rebase。 rebaserebase。

`#`rebase

```
# Rebase 36d15de..612f2f7 onto 36d15de (3 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
```

rebase`git push`。

`git push --force``git push --force-with-lease` 。 。

`git push --force` **-** `--force-with-lease` **-** 。 /`git pull``git fetch`。 。 reflog。 。

Git 1.7.12 root。 。

```
git rebase -i --root
```

。 。 。

git。 /git。

。

。

-
- DB
- ◦
-
  -
  -
  -

- “”。
- -
- “”

```
$ git log --oneline master..
975430b db adding works: db.sql logic.rb
3702650 trying to allow adding todo items: page.html logic.rb
43b075a first draft: page.html and db.sql
$ git rebase -i master
```

```
pick 43b075a first draft: page.html and db.sql
pick 3702650 trying to allow adding todo items: page.html logic.rb
pick 975430b db adding works: db.sql logic.rb
```

```
e 43b075a first draft: page.html and db.sql
e 3702650 trying to allow adding todo items: page.html logic.rb
e 975430b db adding works: db.sql logic.rb
```

git。

```
Stopped at 43b075a92a952faf999e76c4e4d7fa0f44576579... first draft: page.html and db.sql
You can amend the commit now, with

        git commit --amend

Once you are satisfied with your changes, run

        git rebase --continue

$ git status
rebase in progress; onto 4975ae9
You are currently editing a commit while rebasing branch 'feature' on '4975ae9'.
  (use "git commit --amend" to amend the current commit)
  (use "git rebase --continue" once you are satisfied with your changes)

nothing to commit, working directory clean
$ git reset HEAD^ #This 'uncommits' all the changes in this commit.
$ git status -s
 M db.sql
 M page.html
$ git add db.sql  #now we will create the smaller topical commits
$ git commit -m "first draft: db.sql"
$ git add page.html
$ git commit -m "first draft: page.html"
$ git rebase --continue
```

。

```
$ git log --oneline
0309336 db adding works: logic.rb
06f81c9 db adding works: db.sql
3264de2 adding todo items: page.html
675a02b adding todo items: logic.rb
272c674 first draft: page.html
08c275d first draft: db.sql
```

rebase

```
$ git rebase -i master
```

```
pick 08c275d first draft: db.sql
pick 272c674 first draft: page.html
pick 675a02b adding todo items: logic.rb
pick 3264de2 adding todo items: page.html
pick 06f81c9 db adding works: db.sql
pick 0309336 db adding works: logic.rb
```

```
pick 08c275d first draft: db.sql
s 06f81c9 db adding works: db.sql
pick 675a02b adding todo items: logic.rb
s 0309336 db adding works: logic.rb
pick 272c674 first draft: page.html
s  3264de2 adding todo items: page.html
```

git rebase /。 。

rebase

```
$ git log --oneline master..
74bdd5f adding todos: GUI layer
e8d8f7e adding todos: business logic layer
121c578 adding todos: DB layer
```

。 。 git rebase。

git-pullrebase

rebasegit`git pull`rebase。

`.gitconfig` `.gitconfig``.git/config`

```
[branch]
autosetuprebase = always
```

`git config [--global] branch.autosetuprebase always`

`git pull--rebase`

```
[pull]
rebase = true
```

`git config [--global] pull.rebase true`

rebase

pull。 `-x`。

`git rebase -i -x make`

rebase`make`。 `make`git。

rebaseAutostash。 。

Gitrebase。

```
git config --global rebase.autostash    # one time configuration
git rebase @{u}                         # example rebase on upstream branch
```

rebase。 rebase。 。 rebaseautostash。 。 。

https://riptutorial.com/zh-TW/git/topic/355/

36: difftools

Examples

`bcomp.exe`

```
git config --global difftool.bc3.path 'c:\Program Files (x86)\Beyond Compare 3\bcomp.exe'
```

`bc3`

```
git config --global diff.tool bc3
```

KDiff3

`.gitconfig`

```
[merge]
    tool = kdiff3
[mergetool "kdiff3"]
    path = D:/Program Files (x86)/KDiff3/kdiff3.exe
    keepBackup = false
    keepbackup = false
    trustExitCode = false
```

`path`KDiff3

KDiff3diff

```
[diff]
    tool = kdiff3
    guitool = kdiff3
[difftool "kdiff3"]
    path = D:/Program Files (x86)/KDiff3/kdiff3.exe
    cmd = \"D:/Program Files (x86)/KDiff3/kdiff3.exe\" \"$LOCAL\" \"$REMOTE\"
```

IntelliJ IDEWindows

```
[merge]
    tool = intellij
[mergetool "intellij"]
    cmd = cmd \"/C D:\\workspace\\tools\\symlink\\idea\\bin\\idea.bat merge $(cd $(dirname 
"$LOCAL") && pwd)/$(basename "$LOCAL") $(cd $(dirname "$REMOTE") && pwd)/$(basename "$REMOTE") 
$(cd $(dirname "$BASE") && pwd)/$(basename "$BASE") $(cd $(dirname "$MERGED") && 
pwd)/$(basename "$MERGED")\"
    keepBackup = false
    keepbackup = false
    trustExitCode = true
```

`cmd`◦ IDE`Program Files (x86)`

IntelliJ IDEdiffWindows

```
[diff]
    tool = intellij
    guitool = intellij
[difftool "intellij"]
    path = D:/Program Files (x86)/JetBrains/IntelliJ IDEA 2016.2/bin/idea.bat
    cmd = cmd \"/C D:\\workspace\\tools\\symlink\\idea\\bin\\idea.bat diff $(cd $(dirname
"$LOCAL") && pwd)/$(basename "$LOCAL") $(cd $(dirname "$REMOTE") && pwd)/$(basename
"$REMOTE")\"
```

`cmd`。 IDE`Program Files (x86)`

difftools https://riptutorial.com/zh-TW/git/topic/5972/difftools

37:

Examples

GitGit

```
$ git submodule add https://github.com/jquery/jquery.git
```

`.gitmodules`;Git`git submodule update``git submodule update`。

Git

。

```
$ git clone --recursive https://github.com/username/repo.git
```

。 `git submodule update --init --recursive`。

。

```
git submodule update --recursive
```

。

```
git submodule foreach git pull <remote> <branch>
```

`git pull`

```
git submodule foreach git pull
```

。 `git status`。

```
git add <submodule_directory>
git commit
```

`git pull` `git pull --rebase`。 。

```
git submodule foreach git pull --rebase
```

```
git submodule update --remote <submodule_directory>
```

SHA1“gitlink”

。

`git checkout abranch --track origin/abranch, git pull` a

```
git submodule update --remote --recursive
```

SHA1

```
git add .
git commit -m "update submodules"
```

-
  ```
  git submodule -b abranch -- /url/of/submodule/repo
  ```

-
  ```
  cd /path/to/parent/repo
  git config -f .gitmodules submodule.asubmodule.branch abranch
  ```

1.8

`the_submodulethe_submodule`

```
$ git submodule deinit the_submodule
$ git rm the_submodule
```

- `git submodule deinit the_submodule`.git / config`the_submodule` s'。 `git submodule update` `git submodule sync``git submodule foreach`the_submodule 。 。 `git submodule init``git submodule update`。
- `git rm the_submodule`。 `.gitmodules`。 `git rm the_submodule` `git submodule deinit the_submodule` `git submodule deinit the_submodule` .git / config。

1.8

1. `.gitmodules`。
2. `.gitmodules``git add .gitmodules`
3. `.git/config`。
4. `git rm --cached path_to_submodule` 。
5. `rm -rf .git/modules/path_to_submodule`
6. `git commit -m "Removed submodule <name>"`
7.
8. `rm -rf path_to_submodule`

1.8

```
$ git mv old/path/to/module new/path/to/module
```

1.8

1. □□`.gitmodules`□□□□□□□□□□□□□□□□`git add .gitmodules`□□□□□□□。
2. □□□□□□□□□□□□□□□□□□ `mkdir -p new/path/to` □。
3. □□□□□□□□□□□□□□□□ `mv -vi old/path/to/module new/path/to/submodule` □。

4. □□Git□□□□□□ `git add` *new/path* /to □。

5. □□git rm --cached *old/path/to/module*□□□□□git rm --cached *old/path/to/module*。

6. □□□.git/modules/ *old/path/to/module*□□□□□□□.git/modules/ *new/path/to/module*。

7. □□.git/modules/ *new/path/to* /config□□□□□worktree□□□□□□□□□□□□□□□□□worktree = ../../../../../ *old/path/to/module*。□□□□□□□□□□□□□□□□□□.. then□□。 。□□□□*new/path/to/module* /.git □□□□□□□□□□□ □□.git□□□□□□□□□□□□□□□□□□□gitdir: ../../../.git/modules/ *new/path/to/module*。

   git status□□□□□□□

   ```
   # On branch master
   # Changes to be committed:
   #   (use "git reset HEAD <file>..." to unstage)
   #
   #       modified:   .gitmodules
   #       renamed:    old/path/to/submodule -> new/path/to/submodule
   #
   ```

8. □□□□□□□。

□□Axel Beckert□ Stack Overflow□□□□□

□□□□□□□□ https://riptutorial.com/zh-TW/git/topic/306/□□□

□□ 38: □□

□□

- git subtree add -P <prefix> <commit>
  - git subtree add -P <prefix> <repository> <ref>
  - git subtree pull -P <prefix> <repository> <ref>
  - git subtree push -P <prefix> <repository> <ref>
  - git subtree merge -P <prefix> <commit>
  - git subtree split -P <prefix> [OPTIONS] [<commit>]

□□

□□□□submodule□□□□□

Examples

□□□□□□□□□□

□□□□

□□□□□□plugin□□□□□□□□□□□□□

```
git remote add plugin https://path.to/remotes/plugin.git
```

□□□□□□□□□□□□□□□□□plugins/demo。 plugin□□□□□□ master□□□□□□□□□□master□□□

```
git subtree add --prefix=plugins/demo plugin master
```

□□□□□

□□□□□□□□□□□

```
git subtree pull --prefix=plugins/demo plugin master
```

Backport□□□□

1. □□□□□□□□□□□□□□□□□□□□

   ```
   git commit -am "new changes to be backported"
   ```

2. □□□□□□□□□□□□□□□□□□□□□□

   ```
   git checkout -b backport plugin/master
   ```

3. Cherry-pick backports□

   ```
   git cherry-pick -x --strategy=subtree master
   ```

4. □□□□□□□□□□

   ```
   git push plugin backport:master
   ```

[illegible] https://riptutorial.com/zh-TW/git/topic/1634/[illegible]

□□ 39: □□/□□□□□□

□□

- git bisect <subcommand> <options>
- git bisect start <bad> [<good>...]
- git bisect reset
- git bisect good
- git bisect bad

Examples

□□□□□□git bisect□

git bisect□□□□□□□□□□□□□□□□□□□。

□□。 □□□□□□□□□HEAD。

```
# start the git bisect session
$ git bisect start

# give a commit where the bug doesn't exist
$ git bisect good 49c747d

# give a commit where the bug exist
$ git bisect bad HEAD
```

git□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□

```
# tell git the revision is good,
# which means it doesn't contain the bug
$ git bisect good

# if the revision contains the bug,
# then tell git it's bad
$ git bisect bad
```

git□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 git□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□git bisect reset□□□bisect□□□□□HEAD。

```
$ git bisect reset
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
$ git bisect run [script] [arguments]
```

□□[script]□[script]□□□□ [arguments]□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□git bisect good□git bisect bad □□□□□□□□□□□□□。 □□0□□good □□□□1-124,126□127□□□□。 125□□□□□□□□□□□□□□□git bisect skip □。

□□□□□□□□□

□□□□□□□master□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 □□□old-rel □。

Git□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□

```
git bisect start master old-rel
```

□□□□git master□□□□□□□□□□□□□□□□□□□□□ old-rel□□□□□□□□□□。

Git□□□□□□□□□□□□□□□。 □□□□□□□□□□。 □□□□□□□□□□

```
git bisect good
```

□□

```
git bisect bad
```

。 □□□□□□□□□□□□□□□git reset□□□□□□□git□□□□□□。

□□□□□□□□git□□□□□□□□□□。

□□□□bisect□□□□□□

```
git bisect reset
```

□git□□□□□□□□。

□□□□□□/□□□□□□□ https://riptutorial.com/zh-TW/git/topic/3645/□□-□□□□□□

□□ 40: □□□□□□□□

□□

□□□□□□□□□□Git□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□.gitignore □ .git/exclude □ git update-index --assume-unchanged□git update-index --skip-tree □□□□□□Git□□□□□□ □□□□□□□□□□□□□□□□□□ □□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□。

Examples

□□.gitignore□□□□□□□□□

□□□□□□□□□□□□□□□□□.gitignore□□□□Git□□□□□□□□□ – □□□□□□□□□□□Git□□。

□□□□□□□ .gitignore□□□□□□□□□□□□□□□□□□□□□/□□□□□□。 .gitignore□□□□□□□□□□□□□□□□□□□□□□□

1. □□□□□□□□□□□□□□□□□□□□。
2. □□□□□□□□□□□□□□□□□□
3. □□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□

1. □Git□□
2. □git status□git diff□□□□□
3. □□git add -A□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□。□□□□ □□□□□□Git□□□□□□ 。

□□

□□□.gitignore□□□□□glob□□□□□□□□□□□□□□

```
# Lines starting with `#` are comments.

# Ignore files called 'file.ext'
file.ext

# Comments can't be on the same line as rules!
# The following line ignores files called 'file.ext # not a comment'
file.ext # not a comment

# Ignoring files with full path.
# This matches files in the root directory and subdirectories too.
# i.e. otherfile.ext will be ignored anywhere on the tree.
dir/otherdir/file.ext
otherfile.ext

# Ignoring directories
# Both the directory itself and its contents will be ignored.
bin/
gen/

# Glob pattern can also be used here to ignore paths with certain characters.
# For example, the below rule will match both build/ and Build/
```

```
[bB]uild/

# Without the trailing slash, the rule will match a file and/or
# a directory, so the following would ignore both a file named `gen`
# and a directory named `gen`, as well as any contents of that directory
bin
gen

# Ignoring files by extension
# All files with these extensions will be ignored in
# this directory and all its sub-directories.
*.apk
*.class

# It's possible to combine both forms to ignore files with certain
# extensions in certain directories. The following rules would be
# redundant with generic rules defined above.
java/*.apk
gen/*.class

# To ignore files only at the top level directory, but not in its
# subdirectories, prefix the rule with a `/`
/*.apk
/*.class

# To ignore any directories named DirectoryA
# in any depth use ** before DirectoryA
# Do not forget the last /,
# Otherwise it will ignore all files named DirectoryA, rather than directories
**/DirectoryA/
# This would ignore
# DirectoryA/
# DirectoryB/DirectoryA/
# DirectoryC/DirectoryB/DirectoryA/
# It would not ignore a file named DirectoryA, at any level

# To ignore any directory named DirectoryB within a
# directory named DirectoryA with any number of
# directories in between, use ** between the directories
DirectoryA/**/DirectoryB/
# This would ignore
# DirectoryA/DirectoryB/
# DirectoryA/DirectoryQ/DirectoryB/
# DirectoryA/DirectoryQ/DirectoryW/DirectoryB/

# To ignore a set of files, wildcards can be used, as can be seen above.
# A sole '*' will ignore everything in your folder, including your .gitignore file.
# To exclude specific files when using wildcards, negate them.
# So they are excluded from the ignore list:
!.gitignore

# Use the backslash as escape character to ignore files with a hash (#)
# (supported since 1.6.2.1)
\#*#
```

□□□.gitignore□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□.gitignore□□□□□□□□□□□□□□□□□□/□□□□□□。 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□.gitignore

.gitignore□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- □□.git/info/exclude□□□□□□.gitignore□□□□□□。□□□□□□□□□□□□□□□;
- □□□□□□gitignore□□ □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□git□□□

- git update-index --skip-worktree [<file>...] □□□□□□□□□
- git update-index --assume-unchanged [<file>...] □□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□git update-index□□ □□□□□□□□□□□ 。

□□□□□□□□

□□□□□git clean -X□□□□□□□□□□

```
git clean -Xn #display a list of ignored files
git clean -Xf #remove the previously displayed files
```

□□□ -X □□□□ □□□□□□□□□。□□-x □□□□□□□□□□□□□□□□。

□□□□□□□□□□□□git clean□□ 。

□□□□□□□□□□Git□□ 。

.gitignore□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。□□□

```
*.txt
!important.txt
```

□□□□□□□Git□□□□□□□□.txt□□□□□□important.txt□□□□。

□□□□□□□□□□□□□□□□□□□□□□□□□

```
folder/
!folder/*.txt
```

□□□□□□□□□□□□□.txt□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□*□□□□folder□□□□□□□□□□□□folder□*.txt □□□□□□

```
!folder/
folder/*
!folder/*.txt
```

□□ □□□□□□□□□□□□□□□□□□□□□□□□□\□□□□□

```
!!includethis
\!excludethis
```

□□.gitignore□□

□□Git□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□.gitignore □

```
$ git config --global core.excludesfile <Path_To_Global_gitignore_file>
```

□□□□□□□□□□.gitignore□□□□□Git□□□□□□。□□□□

- □□□□.gitignore□□□□□□□□□□□□□□.gitignore□□□□□□□□.gitignore□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□.gigignore□□□□□□□□□□□□□□□□□□□□□repo□□□□□□.gitignore□PC□□□□□□ 。□□□□□□□□□□□□□□□□□□□□□□repo□.gitignore□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□OSX .DS_Store □Windows Thumbs.db□Vim *.ext~□*.ext.swp□□□□□□□□□□□□□□□□□□ 。□□□□□□OS X□□□□□□□□□□□□□□.DS_STORE□_MACOSX □□□□□□□□□Windows□□□□□□□□□□□□□□□thumbs.bd

□□□□□□Git□□□□□□

□□□□□□□□□□□Git□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
git rm --cached <file>
```

□□□□□□□□□□□□□□□□Git□□□□□□□□。 --cached□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□Git□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□ 。

□□□□Git□□□□□□□□□□□□□□□□□□□“ skip worktree ”□□□□□□□□□□□□□□□□□□□

```
git update-index --skip-worktree <file>
```

□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□;□□□□□□□□□□□□□□□□□□□□□□

```
git update-index --no-skip-worktree <file>
```

□□□□□□□□□Git□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
git update-index --assume-unchanged <file>
```

□□□□git□□□ □

□□□□□git□□“□□”□□□□□□□□□□□□□

```
git update-index --no-assume-unchanged <file>
```

□□□□□□□□□

git check-ignore□□□□Git□□□□□。

□□□□□□□□□□□□□□ git check-ignore□□□□□□□□□□。□□□

```
$ cat .gitignore
*.o
$ git check-ignore example.o Readme.md
example.o
```

□□□.gitignore□□□□□* .o□□□□□git check-ignore□□□□□□□□Readme.md。

□□□□□□.gitignore□□□□□□□□□□□git check-ignore□□□□□-v□

```
$ git check-ignore -v example.o Readme.md
.gitignore:1:*.o        example.o
```

□Git 1.7.6□□□□□□□□□git status --ignored□□□□□□□□□。□□□□□□□□□□□.gitignore□□□□□□□□□□□□□□。

□□□□□□□□□□□□□**gitignore**□□□

□□□□□□□□□□□□□□□

```
examples/
    output.log
src/
    <files not shown>
    output.log
README.md
```

examples□□output.log□□□□□□□□□□□□□□□□□src/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□.gitignore□□□□

```
# /.gitignore
src/output.log
```

□□□□□□□src/□□□□□.gitignore□□□□□□□□.gitignore□□□□□□

```
# /src/.gitignore
output.log
```

□□□□□□□□□□

□□□□□□□□□□□foo.txt □□□□□□□□□□□

```
foo.txt # matches all files 'foo.txt' in any directory
```

□□□□□□□□□□□□□□□□□□□□□**□□□□□□□□□□□□□

```
bar/**/foo.txt # matches all files 'foo.txt' in 'bar' and all subdirectories
```

□□□□□□□bar/□□□□□.gitignore□□。□□□□□□□□□□□□□□□□□□□bar/.gitignore □

```
foo.txt # matches all files 'foo.txt' in any directory under bar/
```

□□□□□□□□□□□□□□□□□□□

.gitignore□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□.gitignore □□□□□□□□□□□□□□□□□。

□□□.git/info/exclude 。

□□□

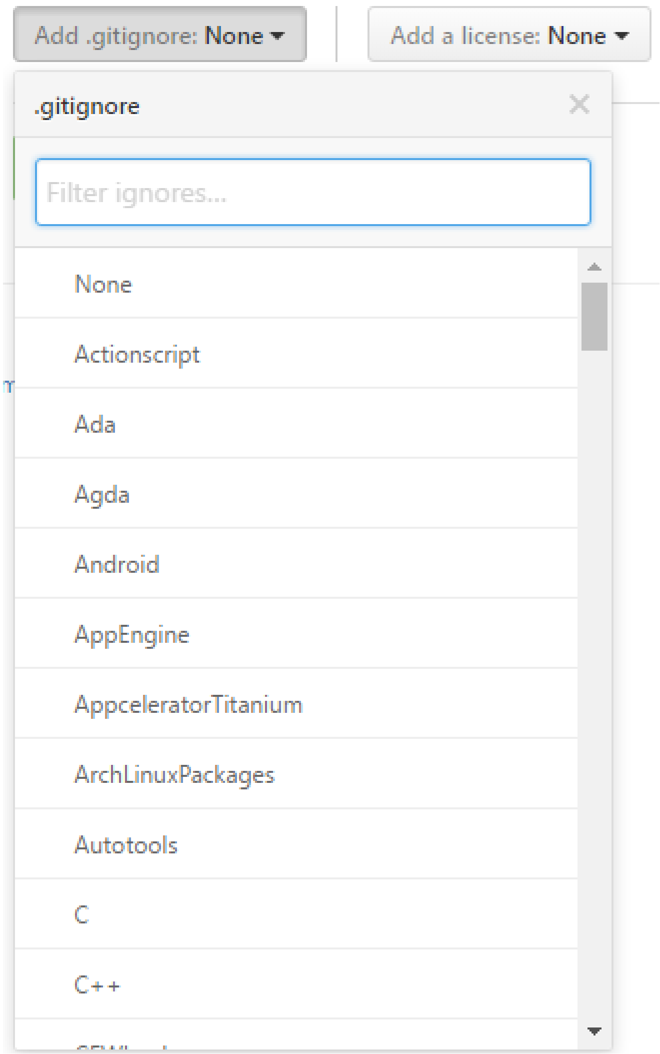
```
# these files are only ignored on this repo
# these rules are not shared with anyone
# as they are personal
gtk_tests.py
gui/gtk/tests/*
localhost
pushReports.py
server/
```

□□□**.gitignore**□□

□□□□□□□□.gitignore□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□.gitignore□□□

- https://www.gitignore.io/
- https://github.com/github/gitignore

□□□□□□□□GitHub□BitBucket□□□□□□□□□□□□□□□□IDE□□.gitignore□□□



□□□□□□□□□□□□□□□□

□□□□□□Git□□□□□□□□□□□□□□□。

□□Git□□update-index□□□□□□□□□□□□

```
git update-index --assume-unchanged my-file.txt
```

□□□□□□□Git□□my-file.txt□□□□□□□□□□□□□□□□。□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□□□□

```
# create a file with some values in
cat <<EOF
MYSQL_USER=app
MYSQL_PASSWORD=FIXME_SECRET_PASSWORD
EOF > .env
```

```
# commit to Git
git add .env
git commit -m "Adding .env template"

# ignore future changes to .env
git update-index --assume-unchanged .env

# update your password
vi .env

# no changes!
git status
```

□□□□□□□□□[stub]

□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□□□.gitignore□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□Git“□□□”□□□。 □□□□□□□□□□。

□□□□□□□file1.c□□□

```
struct settings s;
s.host = "localhost";
s.port = 5653;
s.auth = 1;
s.port = 15653; // NOCOMMIT
s.debug = 1; // NOCOMMIT
s.auth = 0; // NOCOMMIT
```

□□□□□□□□□□□NOCOMMIT□。

□□□□□□□Git□□□□□□.git/config□□□“nocommit”□□□□

```
[filter "nocommit"]
    clean=grep -v NOCOMMIT
```

□□□□□□□□□□.git/info/attributes□.gitmodules □

```
file1.c filter=nocommit
```

□□NOCOMMIT□□□Git□□□□。

□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□Windows□。
- □Git□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□。
- □□Windows□□□□

□□□□□□□□□□。 [□□]

.gitignore□.git/info/exclude□□□□□□□□□□。

□□□□□□□□□□ignore□□□□□□□□update-index □

```
git update-index --skip-worktree myfile.c
```

□□□□□□□□□□□

```
git update-index --no-skip-worktree myfile.c
```

[illegible]git[illegible]git hide [illegible] git unhide[illegible]git hidden[illegible]

```
[alias]
    hide   = update-index --skip-worktree
    unhide = update-index --no-skip-worktree
    hidden  = "!git ls-files -v | grep ^[hsS] | cut -c 3-"
```

[illegible]--assume-unchanged[illegible]update-index[illegible]

```
git update-index --assume-unchanged <file>
```

[illegible]

```
git update-index --no-assume-unchanged <file>
```

[illegible]--assume-unchanged[illegible]Git[illegible]。[illegible]Git[illegible];[illegible]。[illegible]。

[illegible]--skip-worktree[illegible]git[illegible]/[illegible]。[illegible]Skip-worktree[illegible]。

[illegible].gitignore[illegible]

[illegible]git[illegible].gitignore[illegible]。[illegible].gitignore[illegible]。[illegible]。

[illegible]“[illegible]”[illegible]。[illegible]--cached[illegible]

```
# Remove everything from the index (the files will stay in the file system)
$ git rm -r --cached .

# Re-add everything (they'll be added in the current state, changes included)
$ git add .

# Commit, if anything changed. You should see only deletions
$ git commit -m 'Remove all files that are in the .gitignore'

# Update the remote
$ git push origin master
```

[illegible]

[illegible]Git[illegible]Git[illegible]。[illegible]

[illegible] .gitkeep

[illegible].gitkeep[illegible].gitkeep[illegible]Git[illegible]。[illegible].gitkeep[illegible]。[illegible]。[illegible]Windows[illegible]git bash[illegible]

```
$ touch .gitkeep
```

[illegible].gitkeep[illegible]

[illegible] dummy.txt

[illegible]hack[illegible].gitkeep [illegible]dummy.txt。[illegible]Windows[illegible]。

□□□□□□□□□□□□□□。□□□□□□.gitkeep□□□□□□□□。 .gitkeep□□□□□□□□□□□□□□□directoy。

□□.gitignore□□□□□

□□□□□□□□□□□□□□□□git□□□□□□□□

```
git status --ignored
```

□□□□□□□□□□□□□□□□□

```
.git
.gitignore
./example_1
./dir/example_2
./example_2
```

...□.gitignore□□□□□

```
example_2
```

...□□□□□□□□□

```
$ git status --ignored

On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

.gitignore
.example_1

Ignored files:
  (use "git add -f <file>..." to include in what will be committed)

dir/
example_2
```

□□□□□□□□□□□□□□□□□□□□□□□□□ - - --untracked-files=all

□□□□□□□□

```
$ git status --ignored --untracked-files=all
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

.gitignore
example_1

Ignored files:
  (use "git add -f <file>..." to include in what will be committed)

dir/example_2
```

```
example_2
```

□□□□□□□□□□□□□ https://riptutorial.com/zh-TW/git/topic/245/□□□□□□□□

□□ 41: □□

Examples

□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
git reflog
```

□□□□□□□□□□□□□□□□□□□

```
git reset HEAD --hard <sha1-of-commit>
```

□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□ID。

```
git log --diff-filter=D --summary
```

□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□

```
git checkout 81eeccf~1 <your-lost-file-name>
```

□□81eeccf□□□□□□□□□ID□

□□□□□□□□□□□

□□□□□□□□□□□□□□□□□reset。

```
git reset <sha1-of-commit> <file-name>
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□--hard□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
git reflog
```

□□□□□□□□□□□□□□□□

```
git checkout -b <branch-name> <sha1-of-commit>
```

□□git□□□□□□□□□□□□□ – □□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□/□□□□□□□□

□□□□□□

□□**Git**□□□□□□□□□□□□□□□

□□□□□□□□□□*□□□。 □□□□□□Git□□□□□□□□90□□□□□□□□□□□□□reflog□□□□□□

```
$ git reset @~3   # go back 3 commits
$ git reflog
c4f708b HEAD@{0}: reset: moving to @~3
2c52489 HEAD@{1}: commit: more changes
4a5246d HEAD@{2}: commit: make important changes
e8571e4 HEAD@{3}: commit: make some changes
... earlier commits ...
$ git reset 2c52489
... and you're back where you started
```

* □□□*--hard*□*--force*□□□□□ - □□□□□□□□。
* □□□□□□□□□□□□□□□□□□□□□□□。

□git stash□□□

□□□□git stash□□□□□□□□□□□□□

```
git stash apply
```

□□□□□□□□□□□□□□

```
git stash list
```

□□□□□□□□□□□□□

```
stash@{0}: WIP on master: 67a4e01 Merge tests into develop
stash@{1}: WIP on master: 70f0d95 Add user role to localStorage on user login
```

□□□□□□□git□□□□□□□□□□□□□□□□□□

```
git stash apply stash@{2}
```

□□□□□□'git stash pop'□□□'git stash apply'□□□□□..

```
 git stash pop
```

□□

```
 git stash pop stash@{2}
```

git stash apply□git stash pop□□□......

git stash pop □ - □□□□□□□□□□□□□□□。

□□□ -

```
git stash list
```

□□□□□□□□□□□□□

```
stash@{0}: WIP on master: 67a4e01 Merge tests into develop
stash@{1}: WIP on master: 70f0d95 Add user role to localStorage on user login
```

□□□□□□□□□□□□

```
git stash pop
```

[illegible]

```
git stash list
```

[illegible]

```
 stash@{0}: WIP on master: 70f0d95 Add user role to localStorage on user login
```

[illegible]@ {1}[illegible]@ {0}。

[illegible] https://riptutorial.com/zh-TW/git/topic/725/[illegible]

□□ 42: □

□□

□□□Git□□□□□□□□□□□□□□□□□□□□□□□□Git□□□□□□□□□□□□□□□□□□□□□□“□”□□□□□□。 □□□□□□□□Git□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□

- `git pull [options [<repository> [<refspec> ...]]`

□□

--quiet	
-q	--quiet
--verbose	。 fetchmerge / rebase。
-v	--verbose
--[no-]recurse-submodules[=yes | on-demand | no]	/

□□

git pull□□□□□□□□□git fetch□□□git merge□□□□□□□□□□□□□□□□。

Examples

□□□□□□□□□□

□□□□□□□□□ git pull□□□□□□□□

□□□□□□□□□□□□□□□□□□

□□□□□□□svn update□□subversion□□□□□□□□

```
git stash
git pull --rebase
git stash pop
```

□□□□□□□□□□□□□□□□□□□

2.9

```
git config --global alias.up '!git stash && git pull --rebase && git stash pop'
```

2.9

```
git config --global alias.up 'pull --rebase --autostash'
```

□□□□□□□□□□□□

```
git up
```

□□□□□□

```
git pull
```

□□□□□□

```
git fetch
git reset --hard origin/master
```

□□□□□reset --hard□□□□□□□□□reflog□reset□□□ reflog□□□□□□□□□□。

□□□□□□□□□□□origin□master□□□□□□□□□□□□□□。

□□□□□□□□□

□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□git□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□□□□□Git □□□□□□□□□□□□□□。

```
git pull --rebase
```

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□

```
git config branch.autosetuprebase always
```

□□□□□□□□□□□□□□□□□□□

```
git config branch.BRANCH_NAME.rebase true
```

□

```
git pull --no-rebase
```

□□□□□□□□□。

□□□□□□□□

□□□□□□□□□□□□□□□□□□□

```
git pull --ff-only
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□“□□□□□”

□□.git□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□.git□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□.git□□□□□□□□□。

□□□□□□□

```
chown -R youruser:yourgroup .git/
```

□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□GitHub□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

```
git pull
```

□□□□□□□□□□□□。

□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□

```
git pull origin feature-A
```

□□□feature-A□□origin□□□□□□□□。□□□□□□□□□□□URL□□□□□□□□□□□□□□□□□□□SHA□□□□□□□。

□□□□

□□□git pull□□□□□□□□□git fetch□□git merge

```
git fetch origin # retrieve objects and update refs from origin
git merge origin/feature-A # actually perform the merge
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□git branch -a□□□□□□□□□□□□□□□□

```
git checkout -b local-branch-name origin/feature-A # checkout the remote branch
# inspect the branch, make commits, squash, ammend or whatever
git checkout merging-branches # moving to the destination branch
git merge local-branch-name # performing the merge
```

□□□□□□□□□□□□□□。

□□□□□□ https://riptutorial.com/zh-TW/git/topic/1308/□

□□ 43: □□

□□

□□□□□□□□□□□□□□□repo□□□□□□□

```
git bundle create initial.bundle master
git tag -f some_previous_tag master  # so the whole repo does not have to go each time
```

□□□□□□□□□□□□□;□

```
git clone -b master initial.bundle remote_repo_name
```

Examples

□□□□□□□□□git□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□git□□□□□□。 Bundles□□□□□□□□□□□□□□□□git□□□□□□□□□□□□□□□□□□□□□□□□。

```
git tag 2016_07_24
git bundle create changes_between_tags.bundle [some_previous_tag]..2016_07_24
```

□□□□□□**changes_between_tags.bundle**□□□□□□□□□;□□□□□□□□□□。 □□□□□□

```
git bundle verify changes_between_tags.bundle  # make sure bundle arrived intact
git checkout [some branch]       # in the repo on the remote machine
git bundle list-heads changes_between_tags.bundle # list the references in the bundle
git pull changes_between_tags.bundle [reference from the bundle, e.g. last field from the
previous output]
```

□□□□□□□□。 □□□□□□□□□□□□□□□□□□□□;□□git □ ssh □ rsync□http□□。

□□□□□□□ https://riptutorial.com/zh-TW/git/topic/3612/□□

□□ 44: □□□□

□□

Cherry-pick□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□Git SCM Book

□□

- git cherry-pick [- □□] [-n] [-m parent-number] [-s] [-x] [--ff] [-S [key-id]]□□...
- git cherry-pick --continue
- git cherry-pick --quit
- git cherry-pick --abort

□□

-e -	git cherry-pick。
-X	“”。 。
--ff	HEADcherry-pick'ed。
-	.git / sequencer。 。
-	。 。
--abort	。

Examples

□□□□□□□□□□□□□□□□□

git cherry-pick <commit-hash>□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□。

□□□□□□□□□

```
dd2e86 - 946992 - 9143a9 - a6fd86 - 5a6057 [master]
           \
            76cada - 62ecb3 - b886a0 [feature]
```

□□□□□□b886a0□□□master□□5a6057 □。

□□□□□

```
git checkout master
git cherry-pick b886a0
```

□□□□□□□□□□□

```
dd2e86 - 946992 - 9143a9 - a6fd86 - 5a6057 - a66b23 [master]
           \
            76cada - 62ecb3 - b886a0 [feature]
```

□□□□□a66b23□□□□□□□□□DIFF□□□□□□□□b886a0 □□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□b886a0 □□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□

git cherry-pick <commit-A>..<commit-B>□□□□□□□A □□ □□□□□□B□□□□□□□□□□。

git cherry-pick <commit-A>^..<commit-B>□□□A□□□□□□□□□□□□□□□□□□B.

□□□□□□□□□□

□□□。

git branch --contains <commit>□□□□□□□□□□□□□。

git branch -r --contains <commit>□□□□□□□□□□□□□。

□□□□□□□□□□□□

□□git cherry□□□□□□□□□□□。

□□

```
git checkout master
git cherry development
```

...□□□□□□□□□□□

```
+ 492508acab7b454eee8b805f8ba906056eede0ff
- 5ceb5a9077ddb9e78b1e8f24bfc70e674c627949
+ b4459544c000f4d51d1ec23f279d9cdb19c1d32b
+ b6ce3b78e938644a293b2dd2a15b2fecb1b54cd9
```

□+□□□□□□□□□development□□□。

□□□

git cherry [-v] [<upstream> [<head> [<limit>]]]

□□□

-v□□SHA1□□□□□□□。

<upstream>□□□□□□□□□□□□□。□□□HEAD□□□□□。

<head>□□□□;□□□HEAD。

<limit>□□□□□□□□□□□。

□□git-cherry□□□□□□□□。

□□□□□□□□□ https://riptutorial.com/zh-TW/git/topic/672/□□□□

[illegible] 45: [illegible]

[illegible]

[illegible]Git[illegible]。[illegible]Git[illegible]。

[illegible]

- `git push [-f | --force] [-v | --verbose] [<remote> [<refspec> ...]]`

[illegible]

-	。 。
--verbose	。
<>	。
<Refspec> ...	。

[illegible]

[illegible]

> [illegible]"[illegible]"。[illegible]"[illegible]"[illegible]。
>
> [illegible]"[illegible]"[illegible]。[illegible]。[illegible]。
>
> [illegible]"[illegible]"[illegible]。[illegible]。[illegible]"[illegible]"[illegible]。

[illegible]

Examples

[illegible]

```
git push
```

[illegible]。[illegible]Git 2.x[illegible]Git 1.x[illegible]。

[illegible]

[illegible]git[illegible]。[illegible]。

```
git push origin
```

[illegible]

[illegible]feature_x [illegible]

```
git push origin feature_x
```

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□git push□□□□□□□□□□。 □□□□□□□□□□□git□□□□□□□□□□□□□/□□□□

```
git push --set-upstream origin master
```

□□□ master□□□origin□□□□□□。 □□□□□-u□□--set-upstream□□□。

□□□□□□□□

□□□□□□□□□□□□□□□□□□

1. □GitHub□□□□□□□□□□□□
2. □□□□□□□□□□□□□https://github.com/USERNAME/REPO_NAME.git
3. □□□□□□□□□□□□git remote add origin URL
    - □□□□□□□□□□□git remote -v
4. □□git push origin master

□□□□□□□□□GitHub□

□□□□□□□□□□□□□□□□□

□□

□□□□□□□git□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□“□□”□“□□”□□□□□□□□□□□□□□□ 。

□□□□

□□□。

```
git push -f
```

□□

```
git push --force
```

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□ 。 □□。

□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□□□□□□
- □□□。

□□□□□□□□□□□□

□□□□

```
git push <remotename> <object>:<remotebranchname>
```

□

```
git push origin master:wip-yourname
```

□□□□□□□□□wip-yourname□□□□□□□□□□□□□□□□□□□。

□□□□□□

□□□□□□□□□□□□□□□□□。

```
git push <remotename> :<remotebranchname>
```

□

```
git push origin :wip-yourname
```

□□□□□□□wip-yourname

□□□□□□--delete□□□□□□□□□□□□□□□□□□□□□。

□

```
git push origin --delete wip-yourname
```

□□□□□□

□□

```
git push <remotename> <commit SHA>:<remotebranchname>
```

□

□□□□□□git□□

```
eeb32bc Commit 1 - already pushed
347d700 Commit 2 - want to push
e539af8 Commit 3 - only local
5d339db Commit 4 - only local
```

□□□*347d700*□□□□□□□□□□□□□□

```
git push origin 347d700:master
```

□□□□□□□□

Current□□□□□□□□□□□□□□□□□□□□□□。

```
git config push.default current
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□。

```
git config push.default simple
```

□□□□□□□ □□□□□□□□□□□。

```
git config push.default upstream
```

□□□□□□□□□□□□git config push.default□□□□□□□

□□□□□□□□□□□

```
git push
```

□□□□□□□。

□□□

```
git push --tags
```

□□□□□□□□□□□□□□□□□□□git tags。

□□□□□□□ https://riptutorial.com/zh-TW/git/topic/2600/□□

□□ 46: □□

□□

□□Git□□□□□□□□□□□□□□□□□□□□。 Git□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□Git□□□□□□□□□□□□。

□□

- `git commit [flags]`

□□

- -m	。 Git。
-	。
--no	。 `git commit --amend --no-edit`。
--all-a	。
-	。
-	。 。
--patch-p	。
-	`git commit`
-S [keyid] - S --gpg-sign [= keyid] - S --no-gpg-sign	GPGcountermand `commit.gpgSign`
-n - no-verify	pre-commitcommit-msg。

Examples

□□□□□□□□□□□□□

□□□□`git commit`□□Git□□□□□□□□□□□□□vim□emacs □。 □□-m□□□□□□□□□□□□

```
git commit -m "Commit message here"
```

□□□□□□□□□□□□□

```
git commit -m "Commit 'subject line' message here

More detailed description follows here (after a blank line)."
```

□□□□□□□□□□-m□□□

```
git commit -m "Commit summary" -m "More detailed description follows here"
```

□□□*□□□□Git□□□□* 。

Udacity Git□□□□□□□□

□□□□

□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□。

```
git commit --amend
```

□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□vi / vim / emacs □□□□□□□□□□□□。

□□□□□□□□□□

```
git commit --amend -m "New commit message"
```

□□□□□□□□□□□□□□□□□

```
git commit --amend --no-edit
```

□□□□□□□□□□□□□□□□□。 □□□□□git□□□□。

```
git commit --amend --reset-author
```

□□□□□□□□□□□□□□□□□□

```
git commit --amend --author "New Author <email@address.com>"
```

□□。 □□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□□□□□□□push --force。

□□□□□□

□□□□□□□□git add□git rm□□□□□□□□□□□□□□□□git commit。 □□-a□--all□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
git commit -a
```

□□□□□□□□□□□□□□□□□□□□□□

```
git commit -a -m "your commit message goes here"
```

□□□□□□□□□□□□□

```
git commit -am "your commit message goes here"
```

□□□□□□□□□□□□□□。 □□-a□--all□□□□□□□□□□□□□□

```
git commit path/to/a/file -m "your commit message goes here"
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
git commit path/to/a/file path/to/a/folder/* path/to/b/file -m "your commit message goes here"
```

□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□CI□□□□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□。

--allow-empty□□□□□□□。

git commit -m "This is a blank commit" --allow-empty

□□□□□□□

□□

□□□□□□□□□□□□□ Git□□□□□□□□□□□□□□。

□□□□□□□□README.md□program.py □

```
git add README.md program.py
```

□□□git□□□□□□□□□□□□□□□□。

□□□□□□□□□

```
git commit
```

□□□□□□□□□□□□□□□□□□□□ vim。 □□□□□□vim□□□□□□□□□□□□i□□□□□□□□□□□□□□□□□□Esc□:wq□□□□□。 □□□□□□□□□□□□□□□□□□□□-m□□□□

```
git commit -m "Commit message here"
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 。

□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
git add --all        # equivalent to "git add -a"
```

□□□□□□□□□□□□□□□□ □□□□□□□ □

```
git add .
```

□□□□□□□□□□□□“□□”□□

```
git add -u
```

□□□□□□□□□□□□□

```
git status           # display a list of changed files
git diff --cached    # shows staged changes inside staged files
```

□□□□□□□

```
git commit -m "Commit message here"
```

□□git add□git commit□□□□

```
git commit -am "Commit message here"
```

□□□□□□□git add --all□□□□□□git add --all □□□□□□□。

□□□□

□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□。□□□□□□□□□□□□□“BFG Repo-Cleaner”□ https □ //rtyley.github.io/bfg-repo-cleaner/ 。

□□bfg --replace-text passwords.txt my-repo.git□passwords.txt□□□□□□□□□□□□□***REMOVED***。□□□□□□□□□□□□□□□。

□□□□□□

□□□□□□□□□□□□□□□□□□□□--author□□--author□□□□□

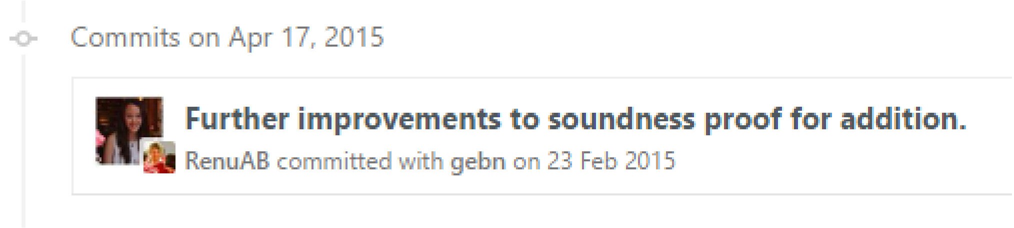
```
git commit -m "msg" --author "John Smith <johnsmith@example.com>"
```

□□□□□□□□□□Git□□□□□□□□□□□□□□

```
git commit -m "msg" --author "John"
```

□□□□□□□□□□□□□□□□□□□□□□□“John”□□□。

□GitHub□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□



Commits on Apr 17, 2015

Further improvements to soundness proof for addition.

RenuAB committed with gebn on 23 Feb 2015

□□□□□□□□

□□□□□□□□□□□□□□□□□git add□□□□□git add □

```
git commit file1.c file2.h
```

□□□□□□□□□□

```
git add file1.c file2.h
```

□□□□□□□□

```
git commit
```

□□□□□

□□git log□□□□□□□□□□□□。□□。

□□□□□□□□□□□

```
TASK-123: Implement login through OAuth
TASK-124: Add auto minification of JS/CSS files
TASK-125: Fix minifier error when name > 200 chars
```

□□□□□□□□□

```
fix                           // What has been fixed?
just a bit of a change        // What has changed?
TASK-371                      // No description at all, reader will need to look at the tracker
themselves for an explanation
Implemented IFoo in IBar      // Why it was needed?
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□___□□□□□□。

□□□□□git□□□□□□□□□

1. □□□□□□□□□□□□
2. □□□□□□□50□□□
3. □□□□□
4. □□□□□□□□□□
5. □□□□□□□□□□□□
6. □□□□□□□□□□□□□72□□□□
7. □□□□□□□□□□□□□□□□□

Chris Beam□□□□□ 7□□□ 。

□□□□□□□

```
git commit -m 'Fix UI bug' --date 2016-07-01
```

--date□□□□□□□□ 。 □□□□□□□□□□git log□□□□□□。

□□□□□□□ □

```
GIT_COMMITTER_DATE=2016-07-01 git commit -m 'Fix UI bug' --date 2016-07-01
```

date□□□□GNU□□□□□□□□□□□□□

```
git commit -m 'Fix UI bug' --date yesterday
git commit -m 'Fix UI bug' --date '3 days ago'
git commit -m 'Fix UI bug' --date '3 hours ago'
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□

□□

```
git add -p
```

□□

```
git add -p [file]
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
y - Yes, add this hunk

n - No, don’t add this hunk

d - No, don’t add this hunk, or any other remaining hunks for this file.
    Useful if you’ve already added what you want to, and want to skip over the rest.

s - Split the hunk into smaller hunks, if possible

e - Manually edit the hunk.  This is probably the most powerful option.
    It will open the hunk in a text editor and you can edit it as needed.
```

[illegible]。[illegible]

```
git commit -m 'Commit Message'
```

[illegible]。[illegible]

```
git reset --hard
```

[illegible]。[illegible]println / logging[illegible]。

[illegible]

[illegible]

```
git commit --amend --date="Thu Jul 28 11:30 2016 -0400"
```

[illegible]

```
git commit --amend --date="now"
```

[illegible]

[illegible]

```
git config user.name "Full Name"
git config user.email "email@example.com"

git commit --amend --reset-author
```

GPG[illegible]

1. [illegible]ID

   ```
   gpg --list-secret-keys --keyid-format LONG

   /Users/davidcondrey/.gnupg/secring.gpg
   ---------------------------------------
   sec   2048R/YOUR-16-DIGIT-KEY-ID YYYY-MM-DD [expires: YYYY-MM-DD]
   ```

   [illegible]ID[illegible]16[illegible]。

2. [illegible]git[illegible]ID

```
git config --global user.signingkey YOUR-16-DIGIT-KEY-ID
```

3. □□□1.7.9□□□git commit□□-S□□□□□□□□□□□□。 □□□□□□□□□□□GPG□□□□□□□□□□□□□□□□□。

```
git commit -S -m "Your commit message"
```

□□□□□□□ https://riptutorial.com/zh-TW/git/topic/323/□□

□□ 47: □□

Examples

□□□□

□□□□□□□□□□□□

□□。

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□SHA□□□□□□□□□□□□git log□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□。

```
> git reset --hard <last commit from the branch you are on>
```

□□□□□□□□□□□□□□。

```
> git reset HEAD~
```

□□□。

□□□□□□□□□

□□□□□□□□□□□□add-gremlins□

```
> git merge feature/add-gremlins
...
   #Resolve any merge conflicts
> git commit #commit the merge
...
> git push
...
   501b75d..17a51fd  master -> master
```

□□□。

```
> git revert -m 1 17a51fd
...
> git push
...
   17a51fd..e443799  master -> master
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□。 □□add-gremlins□□□□□□□□□□□□□□□□□□□□□□□。

```
> git checkout feature/add-gremlins
...
   #Various commits to fix the bug.
> git checkout master
...
> git revert e443799
...
> git merge feature/add-gremlins
...
   #Fix any merge conflicts introduced by the bug fix
> git commit #commit the merge
...
> git push
```

[illegible]。[illegible]。

```
> git checkout feature/add-gremlins
...
   #Merge in master and revert the revert right away.  This puts your branch in
   #the same broken state that master was in before.
> git merge master
...
> git revert e443799
...
   #Now go ahead and fix the bug (various commits go here)
> git checkout master
...
   #Don't need to revert the revert at this point since it was done earlier
> git merge feature/add-gremlins
...
   #Fix any merge conflicts introduced by the bug fix
> git commit #commit the merge
...
> git push
```

[illegible]reflog

[illegible]rebase[illegible]commit[illegible]pre rebase[illegible]。[illegible]reflog[illegible]90[illegible] – [illegible]

```
$ git reflog
4a5cbb3 HEAD@{0}: rebase finished: returning to refs/heads/foo
4a5cbb3 HEAD@{1}: rebase: fixed such and such
904f7f0 HEAD@{2}: rebase: checkout upstream/master
3cbe20a HEAD@{3}: commit: fixed such and such
...
```

[illegible]rebase[illegible]HEAD@{3}[illegible]

```
git checkout HEAD@{3}
```

[illegible]/[illegible]/[illegible]*rebase*。

[illegible]reflog[illegible]100[illegible]

`git reset --hard HEAD@{3}`

[illegible]git[illegible]。

[illegible]。

[illegible]

[illegible]git log[illegible]。

[illegible]

```
git checkout 789abcd
```

[illegible]789abcd。[illegible]。[illegible]branch[illegible]checkout -b[illegible]。

[illegible]

```
git reset --soft 789abcd
```

□□□□□□□□□□

```
git reset --soft HEAD~
```

□□□□□□□□□□□□□□□□□□□□□□

```
git reset --hard 789abcd
```

□□□□□□□□□□□□□□□□□

```
git reset --hard HEAD~
```

□□□□□□□□□□reflog□reset□□□□□□□ □□□□□□□□□□□□。□□git stash; git reset□□□git reset --hard□□□□。

□□□□

□□□□□□□□□□□□□□□□□。

```
git checkout -- file.txt
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

```
git checkout -- .
```

□□□□□□□□□□□□--patch。 □□□□□□□□□□□□□□□□□□□。

```
git checkout --patch -- dir
```

□□□□□□□□□□。

```
git reset --hard
```

□□--hard□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□。

```
git reset HEAD~2
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□git stash -p□git stash。 □□□□□□□stash pop□□□□□stash drop□□□□stash drop。

□□□□□□□□□

□□git revert□□□□□□□□□□□□□□□□□□□□□□□□□□。 □□。

□□□□□□□□□□□□□□□□□□□□□□□□□□□□git push --force。 □□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
git revert HEAD~1
git push
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
git revert HEAD~1
work .. work .. work ..
git add -A .
git commit -m "Update error code"
git push
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 Git□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

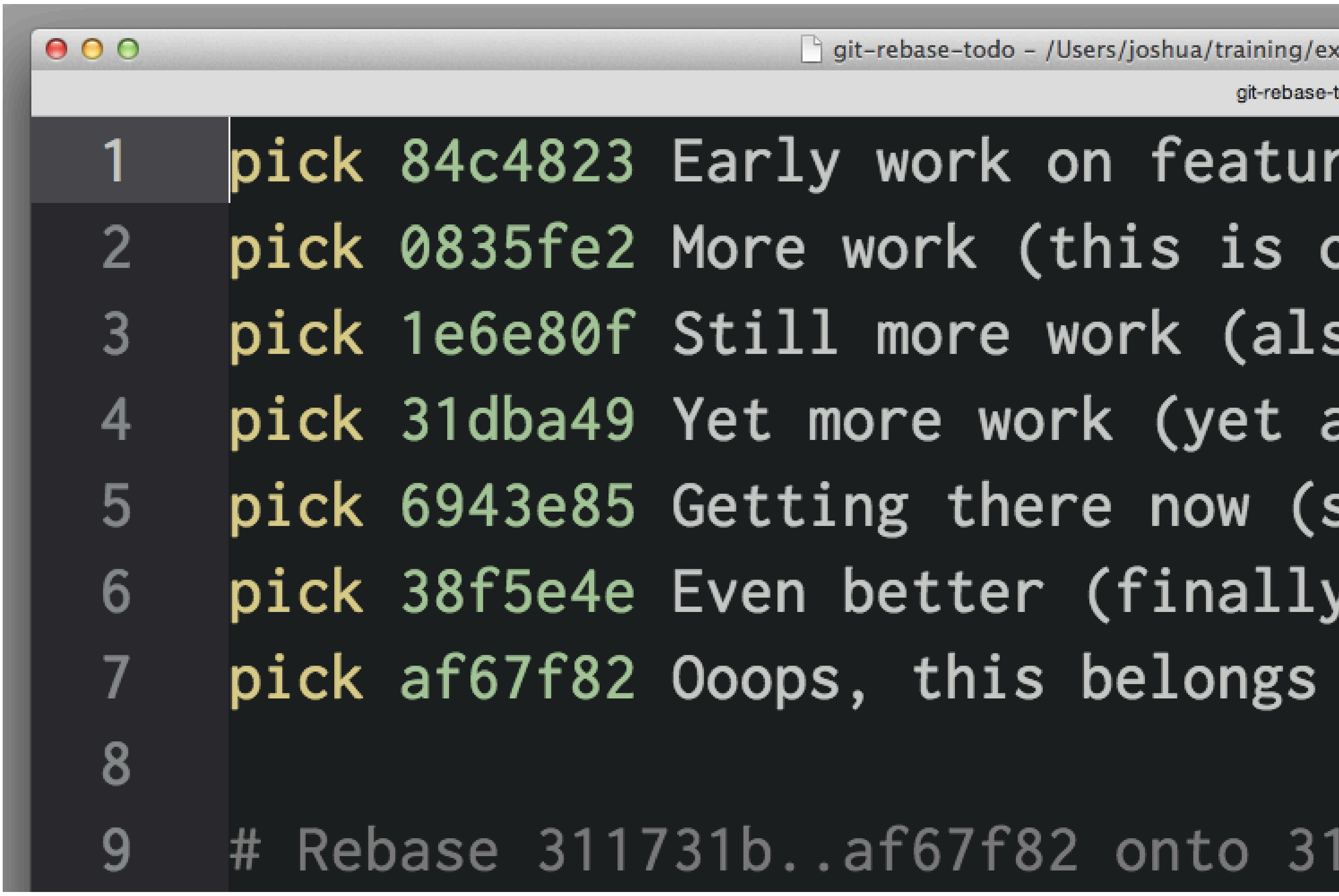
```
git revert 912aaf0228338d0c8fb8cca0a064b0161a451fdc
git push
```

□□/□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□。

```
git rebase -i <earlier SHA>
```

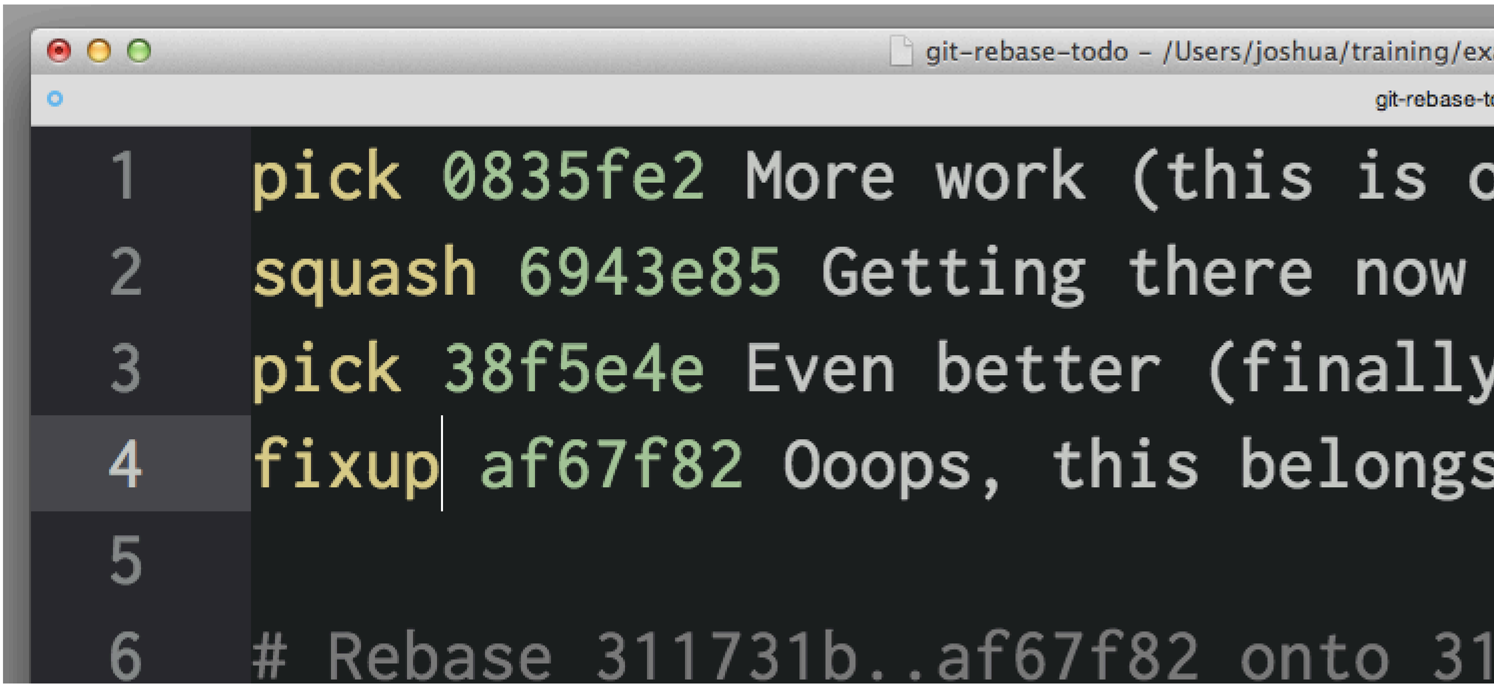
-i□rebase□□“□□□□”。 □□□□□□□rebase□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 *rebase -i*□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□



```
pick 84c4823 Early work on featur
pick 0835fe2 More work (this is c
pick 1e6e80f Still more work (als
pick 31dba49 Yet more work (yet a
pick 6943e85 Getting there now (s
pick 38f5e4e Even better (finally
pick af67f82 Ooops, this belongs

# Rebase 311731b..af67f82 onto 31
```

□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□□□□□□□□□1□□□3-4□。 □□□□□□□□□□□□□□□□□□□squash□fixup□□



```
pick 0835fe2 More work (this is c
squash 6943e85 Getting there now
pick 38f5e4e Even better (finally
fixup af67f82 Ooops, this belongs

# Rebase 311731b..af67f82 onto 31
```

□□□□□□□ https://riptutorial.com/zh-TW/git/topic/285/□□

□□ 48: □□

□□

□□□□□□

Squashing□□□。

□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□;□□□git push -f□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□GitHub□□□□□□□□□□□□□□“□□□□□□”□□□master □□□□□□Settings - Branches - Protected Branches。

Examples

□□□□□□□□□□□□□

□□□□□□□x□□□□□□□□□□□□□□□□□□□□

```
git reset --soft HEAD~x
git commit
```

□x□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□□□□□□x□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□□。

□**Rebase**□□□□□□

□□□□□git rebase□□□□□。 □□□□□□□□□□□□□□□□□□□□□。

1. □□□□□□□□□□□□□□□□□□□□□。
2. □□git rebase -i [commit hash]。

   □□□□□□□□HEAD~4□□□□□□□□□□□□□□□□□□□□□□□□□4□□□。
3. □□□□□□□□□□□□□□□□□□□□□□□□。 □squash□□□□□□□□□□pick □□□□squash□□□□□□□。
4. □□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□

```
> git log --oneline
612f2f7 This commit should not be squashed
d84b05d This commit should be squashed
ac60234 Yet another commit
36d15de Rebase from here
17692d1 Did some more stuff
e647334 Another Commit
2e30df6 Initial commit

> git rebase -i 36d15de
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 Git□□□□□□□□。 □□rebase□□□□□□□。 □□□□□□□□□□□□□□□□

```
pick ac60234 Yet another commit
squash d84b05d This commit should be squashed
pick 612f2f7 This commit should not be squashed

# Rebase 36d15de..612f2f7 onto 36d15de (3 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
```

□□□□□□□Git□□

```
> git log --oneline
77393eb This commit should not be squashed
e090a8c Yet another commit
36d15de Rebase from here
17692d1 Did some more stuff
e647334 Another Commit
2e30df6 Initial commit
```

Autosquash□□□□□□□rebase□□□□□□□

□□□□□□□□□□□□□□□□□□□□□bbb2222 A second commit□□□bbb2222 A second commit □

```
$ git log --oneline --decorate
ccc3333 (HEAD -> master) A third commit
bbb2222 A second commit
aaa1111 A first commit
9999999 Initial commit
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□--fixup□□□□□□□□□□□--fixup□□□□

```
$ git add .
$ git commit --fixup bbb2222
[my-feature-branch ddd4444] fixup! A second commit
```

□□□□□□□□□□□□□□□Git□□□□rebase□□□□□□□□□□□□

```
$ git log --oneline --decorate
ddd4444 (HEAD -> master) fixup! A second commit
ccc3333 A third commit
bbb2222 A second commit
aaa1111 A first commit
9999999 Initial commit
```

□□□□□□--autosquash□□□□□□□rebase□

```
$ git rebase --autosquash --interactive HEAD~4
```

Git□□□□□□□commit --fixup□□□commit --fixup□□□□□□□

```
pick aaa1111 A first commit
pick bbb2222 A second commit
fixup ddd4444 fixup! A second commit
pick ccc3333 A third commit
```

□□□□□□□□rebase□□□--autosquash □□□□□□□□□□□□

```
$ git config --global rebase.autosquash true
```

□□□□□□□□□

□□□□□git merge --squash□□□□□□□git merge --squash□□□□□□。□□□□□□□□□。

```
git merge --squash <branch>
git commit
```

□□□□□□□□□□git reset □□□□□□□□□□□□□□□□□□□。□□□

```
git checkout <branch>
git reset --soft $(git merge-base master <branch>)
git commit
```

□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

git commit --squash=[commit hash of commit to which this commit will be squashed to]

□□□□□--fixup=[commit hash]□--fixup=[commit hash]。

□□□□□□□□□□□□□□□□□□□□□□□□□

git commit --squash :/things

□□□□□“□□”□□□□□□□。

□□□□□□□□□'fixup!'□□'fixup!'□'squash!'□□□□□□□□□□□□□□□□□□□□□□。

□□□--autosquash□□□□□□□□autosquash / fixup□□。

□□□□□□□ https://riptutorial.com/zh-TW/git/topic/598/□□

□□ 49: □□□□□□□□□□□□

Examples

□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□

```
git fetch -p
```

□□□□□□□

```
git branch -vv
```

□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□“□□□”

```
 branch                 12345e6 [origin/branch: gone] Fixed bug
```

□□□□□□□□□□□□□□□□□'git branch -vv'□□'□□'□□□□'-d'□□□□□□□

```
git fetch -p && git branch -vv | awk '/: gone]/{print $1}' | xargs git branch -d
```

□□□□□□□□□□□□□□□□ https://riptutorial.com/zh-TW/git/topic/10934/□□□□□□□□□□□□

□□ 50: □□git□□□□□

□□

□□□□□□□□□□□□□□□□github□bitbucket□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□**。

Examples

□□□□□□

□□□□

```
cd projectFolder
git remote -v (it will show previous git url)
git remote set-url origin https://username@bitbucket.org/username/newName.git
git remote -v (double check, it will show new git url)
git push (do whatever you want.)
```

□□□□□□git□□□□□□ https://riptutorial.com/zh-TW/git/topic/9291/□□git□□□□□

□□ 51: □□

□□

- `git archive [--format = <fmt>] [--list] [--prefix = <prefix> /] [<extra>] [-o <file> | --output = <file>] [ - worktree-attributes] [--remote = <repo> [--exec = <git-upload-archive>]] <tree-ish> [<path> ...]`

□□

--format = <FMT>	tarzip。 。 tar。
-l - list	。
-v - verbose	stderr。
= <> /	<prefix> /。
-o <file> - output = <file>	<file>stdout。
--worktree	.gitattributes。
<>	。 zip-0-1-9。
--remote = <>	<repo>tar。
--exec = <GIT>	--remote<git-upload-archive。
<>	。
<>	。 。

Examples

□□□□□□□□**git**□□□□□□

□□□git□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□HEAD□□□

```
git archive --output=archive-HEAD.zip --prefix=src-directory-name HEAD
```

□□□□□□□□□□□□□□src-directory-name□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□**git**□□□□□□

□□□□□□HEAD□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□dev□□□□

```
git archive --output=archive-dev.zip --prefix=src-directory-name dev
```

□□□□□□□origin/dev□□□□□□□□□□□□

```
git archive --output=archive-dev.zip --prefix=src-directory-name origin/dev
```

□□□□□v.01□□□□

```
git archive --output=archive-v.01.zip --prefix=src-directory-name v.01
```

□□□□HEAD□□□□□□□ sub-dir □□□□□□□□□

```
git archive zip --output=archive-sub-dir.zip --prefix=src-directory-name HEAD:sub-dir/
```

□□**git**□□□□□□

□□git archive □□□□□□□□□□□□□□□□□□□□□□。

□□□□HEAD□□□□tar□□□

```
git archive --format tar HEAD | cat > archive-HEAD.tar
```

□□gzip□□□□□□HEAD□□□□tar□□□

```
git archive --format tar HEAD | gzip > archive-HEAD.tar.gz
```

□□□□□□□□□□□□tar.gz□□□□

```
git archive --format tar.gz HEAD > archive-HEAD.tar.gz
```

□□□□HEAD□□□□zip□□□

```
git archive --format zip HEAD > archive-HEAD.zip
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
git archive --output=archive-HEAD.tar.gz HEAD
```

□□□□□□□ https://riptutorial.com/zh-TW/git/topic/2815/□□

□□ 52: □□

□□

- `git blame [□□□]`
- `git blame [-f] [ - e] [ - w] [filename]`
- `git blame [-L range] [□□□]`

□□

-F	
-e	
-w	
-L	`git blame -L 1,2 [filename]`
--show-	blame
-l	
-t	
-	
-p -	
-M	
-C	-M
-H	
-C	git-annotate
-n	

□□

□□□□□□□□□□□□□□□□□□□□git blame□□□□□□。

Examples

□□□□□□□□□□

```
git blame <file>
```

□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□

□□repos□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□。

```
git blame -w
```

□□□□□□□□□□□□□□□□□□□□。

□□□□□□

□□□□□□□□□□□□□□□□

```
git blame -L <start>,<end>
```

<start>□<end>□□□□

- □□□□

  ```
  git blame -L 10,30
  ```

- /□□□□□/

  `git blame -L /void main/` □ `git blame -L 46,/void foo/`

- + offset□-offset□□□□□<end> □

  `git blame -L 108,+30` □ `git blame -L 215,-15`

□□□□□□□□□□□□□□□□□。

```
git blame -L 10,30 -L 12,80 -L 120,+10 -L ^/void main/,+40
```

□□□□□□□□

```
// Shows the author and commit per line of specified file
git blame test.c

// Shows the author email and commit per line of specified
git blame -e test.c file

// Limits the selection of lines by specified range
git blame -L 1,10 test.c
```

□□□□□□□ https://riptutorial.com/zh-TW/git/topic/3663/□□

□□ 53: □□□□□□

□□

- `git log [options] [revision range] [[ − ] path ...]`

□□

-q - quiet	
-	
--use-mailmap	
--decorate [= ...]	
--L <nmfile>	1。 nm。 。
--show	
-i - rengexp-ignore-case	

□□

□□□□□□□□□ □ git-log□□□□

Examples

“□□”Git□□

```
git log
```

□□□□□□□□□□□□□。 □□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□onelineing □。 □□q□□□□□。

> □□□□□□□□□□□git log□□□□□□□□□□□□□□□□□□□□ − □□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□SHA−1□□□□□□□□□ □□□□□□□□□□□□□□□。 -□□

□□□□□**Free Code Camp**□□□□□

```
commit 87ef97f59e2a2f4dc425982f76f14a57d0900bcf
Merge: e50ff0d eb8b729
Author: Brian <sludge256@users.noreply.github.com>
Date:   Thu Mar 24 15:52:07 2016 -0700

    Merge pull request #7724 from BKinahan/fix/where-art-thou

    Fix 'its' typo in Where Art Thou description

commit eb8b7298d516ea20a4aadb9797c7b6fd5af27ea5
Author: BKinahan <b.kinahan@gmail.com>
Date:   Thu Mar 24 21:11:36 2016 +0000

    Fix 'its' typo in Where Art Thou description
```

```
commit e50ff0d249705f41f55cd435f317dcfd02590ee7
Merge: 6b01875 2652d04
Author: Mrugesh Mohapatra <raisedadead@users.noreply.github.com>
Date:   Thu Mar 24 14:26:04 2016 +0530

    Merge pull request #7718 from deathsythe47/fix/unnecessary-comma

    Remove unnecessary comma from CONTRIBUTING.md
```

□□□□□□□□□□□□□n□□□□□□□□□□□□□□。 □□□□□□□□□□□□2□□□□□

```
git log -2
```

Oneline□□

```
git log --oneline
```

□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□oneline□□□□。

> oneline□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。 -□□

□□□□□**Free Code Camp**□□□□□□□□□□□□□□□□□□

```
87ef97f Merge pull request #7724 from BKinahan/fix/where-art-thou
eb8b729 Fix 'its' typo in Where Art Thou description
e50ff0d Merge pull request #7718 from deathsythe47/fix/unnecessary-comma
2652d04 Remove unnecessary comma from CONTRIBUTING.md
6b01875 Merge pull request #7667 from zerkms/patch-1
766f088 Fixed assignment operator terminology
d1e2468 Merge pull request #7690 from BKinahan/fix/unsubscribe-crash
bed9de2 Merge pull request #7657 from Rafase282/fix/
```

□□□□□□□□□□□□□n□□□□□□□□□□□□□□□。 □□□□□□□□□□□□2□□□□□

```
git log -2 --oneline
```

□□□□□□

□□□□□□□□□□□□□□□□□□□□

```
git log --decorate --oneline --graph
```

□□□□□

```
* e0c1cea (HEAD -> maint, tag: v2.9.3, origin/maint) Git 2.9.3
*   9b601ea Merge branch 'jk/difftool-in-subdir' into maint
|\
| * 32b8c58 difftool: use Git::* functions instead of passing around state
| * 98f917e difftool: avoid $GIT_DIR and $GIT_WORK_TREE
| * 9ec26e7 difftool: fix argument handling in subdirs
* |   f4fd627 Merge branch 'jk/reset-ident-time-per-commit' into maint
...
```

□□□□□□□□□□□□□□□□□□□□□□□

```
git config --global alias.lol "log --decorate --oneline --graph"
```

□□□□□□□□

```
# history of current branch :
git lol

# combined history of active branch (HEAD), develop and origin/master branches :
git lol HEAD develop origin/master

# combined history of everything in your repo :
git lol --all
```

□□□□□□

□□□□□□□□□□□□□□-p□--patch□□。

```
git log --patch
```

□□□□□Trello Scientist□□□□

```
ommit 8ea1452aca481a837d9504f1b2c77ad013367d25
Author: Raymond Chou <info@raychou.io>
Date:   Wed Mar 2 10:35:25 2016 -0800

    fix readme error link

diff --git a/README.md b/README.md
index 1120a00..9bef0ce 100644
--- a/README.md
+++ b/README.md
@@ -134,7 +134,7 @@ the control function threw, but *after* testing the other functions and
readying
 the logging. The criteria for matching errors is based on the constructor and
 message.

-You can find this full example at [examples/errors.js](examples/error.js).
+You can find this full example at [examples/errors.js](examples/errors.js).

 ## Asynchronous behaviors


commit d3178a22716cc35b6a2bdd679a7ec24bc8c63ffa
:
```

□□□□

```
git log -S"#define SAMPLES"
```

□□□□□□□□□□□□□□□REGEXP□□□□□□。□□□□□□□□□□□□□□□□/□□□□□#define SAMPLES。□□□

```
+#define SAMPLES  100000
```

□□

```
-#define SAMPLES  100000
```

```
git log -G"#define SAMPLES"
```

□□□□□□□□□□□□□□□□ □ □□ □□□□□□REGEXP。□□□

```
-#define SAMPLES  100000
+#define SAMPLES  100000000
```

□□□□□□□□□□□□□□

git shortlog□□□□□git log□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

```
$ git shortlog
Committer 1 (<number_of_commits>):
    Commit Message 1
    Commit Message 2
    ...
Committer 2 (<number_of_commits>):
    Commit Message 1
    Commit Message 2
    ...
```

□□□□□□□□□□□□□□□□□□□□□□□□□□

-s

--summary

```
$ git shortlog -s
<number_of_commits> Committer 1
<number_of_commits> Committer 2
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

-n

--numbered

□□□□□□□□□□□□□□□□□□□□□□

-e

--email

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--format

□□□□git log□--format□□□□□□□□□□。

□□□□□□□□□□□□□□□□ 。

□□□□

```
git log --after '3 days ago'
```

□□□□□□□□

```
git log --after 2016-05-01
```

□□□□□□□□□□□□□□□□□□□□□□□□□□GNU□□□□□□□□□□。

□□□□--after□--since。

□□converse□□□□□ - --before□--until。

□□□□□author□□□□。 □□

```
git log --author=author
```

□□□□□□□□□

```
$ git log -L 1,20:index.html
commit 6a57fde739de66293231f6204cbd8b2feca3a869
Author: John Doe <john@doe.com>
Date:   Tue Mar 22 16:33:42 2016 -0500

    commit message

diff --git a/index.html b/index.html
--- a/index.html
+++ b/index.html
@@ -1,17 +1,20 @@
 <!DOCTYPE HTML>
 <html>
-       <head>
-        <meta charset="utf-8">
+
+<head>
+    <meta charset="utf-8">
     <meta http-equiv="X-UA-Compatible" content="IE=edge">
     <meta name="viewport" content="width=device-width, initial-scale=1">
```

□□□□

```
git log --graph --pretty=format:'%C(red)%h%Creset -%C(yellow)%d%Creset %s %C(green)(%cr)
%C(yellow)<%an>%Creset'
```

format□□□□□□□□□□□□□□□

□□	□□
%C(color_name)	□□□□□□□□□
%h□□H	□□□□□□□□□H□□□□□□□
%Creset	□□□□□□□□□□□
%d	□□□□
%s	□□[□□□□]
%cr	□□□□□□□□□□□
%an	□□□□

[illegible]

```
tree = log --oneline --decorate --source --pretty=format:'"%Cblue %h %Cgreen %ar %Cblue %an
%C(yellow) %d %Creset %s"'  --all --graph
```

[illegible]

```
*    40554ac  3 months ago  Alexander Zolotov    Merge pull request #95 from
gmandnepr/external_plugins
|\
| *  e509f61  3 months ago  Ievgen Degtiarenko    Documenting new property
| *  46d4cb6  3 months ago  Ievgen Degtiarenko    Running idea with external plugins
| *  6253da4  3 months ago  Ievgen Degtiarenko    Resolve external plugin classes
| *  9fdb4e7  3 months ago  Ievgen Degtiarenko    Keep original artifact name as this may be
important for intellij
| *  22e82e4  3 months ago  Ievgen Degtiarenko    Declaring external plugin in intellij
section
|/
*  bc3d2cb  3 months ago  Alexander Zolotov    Ignore DTD in plugin.xml
```

[illegible]**Git**[illegible]

git log master..foo[illegible]foo[illegible]master[illegible]。 [illegible]

[illegible]

```
git log --stat
```

[illegible]

```
commit 4ded994d7fc501451fa6e233361887a2365b91d1
Author: Manassés Souza <manasses.inatel@gmail.com>
Date:   Mon Jun 6 21:32:30 2016 -0300

    MercadoLibre java-sdk dependency

 mltracking-poc/.gitignore |  1 +
 mltracking-poc/pom.xml    | 14 ++++++++++++--
 2 files changed, 13 insertions(+), 2 deletions(-)

commit 506fff56190f75bc051248770fb0bcd976e3f9a5
Author: Manassés Souza <manasses.inatel@gmail.com>
Date:   Sat Jun 4 12:35:16 2016 -0300

    [manasses] generated by SpringBoot initializr

 .gitignore                                                                            |  42
++++++++++++
 mltracking-poc/mvnw                                                                   | 233
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 mltracking-poc/mvnw.cmd                                                               | 145
+++++++++++++++++++++++++++++++++++++++
 mltracking-poc/pom.xml                                                                |  74
++++++++++++++++++++
 mltracking-poc/src/main/java/br/com/mls/mltracking/MltrackingPocApplication.java      |  12
++++
 mltracking-poc/src/main/resources/application.properties                              |   0
 mltracking-poc/src/test/java/br/com/mls/mltracking/MltrackingPocApplicationTests.java |  18
```

```
+++++
 7 files changed, 524 insertions(+)
```

□□□□□□□□□

□□git show□□□□□□□□□□

```
git show 48c83b3
git show 48c83b3690dfc7b0e622fd220f8f37c26a77c934
```

□

```
commit 48c83b3690dfc7b0e622fd220f8f37c26a77c934
Author: Matt Clark <mrclark32493@gmail.com>
Date:   Wed May 4 18:26:40 2016 -0400

    The commit message will be shown here.

diff --git a/src/main/java/org/jdm/api/jenkins/BuildStatus.java
b/src/main/java/org/jdm/api/jenkins/BuildStatus.java
index 0b57e4a..fa8e6a5 100755
--- a/src/main/java/org/jdm/api/jenkins/BuildStatus.java
+++ b/src/main/java/org/jdm/api/jenkins/BuildStatus.java
@@ -50,7 +50,7 @@ public enum BuildStatus {

                        colorMap.put(BuildStatus.UNSTABLE, Color.decode( "#FFFF55" ));
-                       colorMap.put(BuildStatus.SUCCESS,  Color.decode( "#55FF55" ));
+                       colorMap.put(BuildStatus.SUCCESS,  Color.decode( "#33CC33" ));
                        colorMap.put(BuildStatus.BUILDING, Color.decode( "#5555FF" ));
```

□**git log**□□□□□□□□

□□□□□□□□□□□□□git□□□

```
git log [options] --grep "search_string"
```

□□

```
git log --all --grep "removed file"
```

□□□□□□□ □□□□□□□□removed file□□□。

□git --invert-grep□□□□□□□--invert-grep□□□□□□。

□□

```
git log --grep="add file" --invert-grep
```

□□□□□□□□add file。

□□□□□□□□□□□ https://riptutorial.com/zh-TW/git/topic/240/□□□□□□

□□ 54: □□

□□

- `git stash list [<options>]`
  - `git stash show [<stash>]`
  - `git stash drop [-q|--quiet] [<stash>]`
  - `git stash ( pop | apply ) [--index] [-q|--quiet] [<stash>]`
  - `git stash branch <branchname> [<stash>]`
  - `git stash [save [-p|--patch] [-k|--[no-]keep-index] [-q|--quiet] [-u|--include-untracked] [-a|--all] [<message>]]`
  - `git stash clear`
  - `git stash create [<message>]`
  - `git stash store [-m|--message <message>] [-q|--quiet] <commit>`

□□

	。 <stash>。
	。 @ {0}@ {1}。
	。
	pop。
	。 。
	。 <stash><stash>。 stash @ {0}<stash>stash @ {<revision>}。
	ref。 。 ;“”。
	stash refgit stash createreflog。 。 ;“”。

□□

Stashing□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□/□□□□□。

Examples

□□□Stashing□

□□□□□□□□□□master□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□□□□□□□。 □□□git stash□□□□□□□□。

□□□□□□git status□□□□□□□□□□

```
(master) $ 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:   business/com/test/core/actions/Photo.c

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

□□□□git stash□□□□□□□□□□□

```
(master) $ git stash
Saved working directory and index state WIP on master:
2f2a6e1 Merge pull request #1 from test/test-branch
HEAD is now at 2f2a6e1 Merge pull request #1 from test/test-branch
```

□□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□。

```
(master) $ git stash
Saved working directory and index state WIP on master:
(master) $ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        NewPhoto.c

nothing added to commit but untracked files present (use "git add" to track)
(master) $ git stage NewPhoto.c
(master) $ git stash
Saved working directory and index state WIP on master:
(master) $ git status
On branch master
nothing to commit, working tree clean
(master) $
```

□□□□□□□□□□□□□□□□□□□。□□□□□□□□□git status□□□□□□

```
(master) $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```

□□□□□□□□□□□□□□git stash apply □□□□□□□□□□□□□□□git stash pop□□□□□□□git stash pop □□

```
(master) $ git stash apply
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:   business/com/test/core/actions/Photo.c

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

□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□**master**。□□□□□□□**dev**□□□ **dev** □□□□git stash apply□□□□□□□□□□**dev**□□□。

```
(master) $ git checkout -b dev
Switched to a new branch 'dev'
(dev) $ git stash apply
On branch dev
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:   business/com/test/core/actions/Photo.c

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

[illegible]

[illegible]。

```
git stash
```

[illegible]`--include-untracked`[illegible]`-u`[illegible]。

```
git stash --include-untracked
```

[illegible]

```
git stash save "<whatever message>"
```

[illegible]`--keep-index`[illegible]`-k`[illegible]。

```
git stash --keep-index
```

[illegible]

```
git stash list
```

[illegible]。
[illegible]

```
stash@{0}: WIP on master: 67a4e01 Merge tests into develop
stash@{1}: WIP on master: 70f0d95 Add user role to localStorage on user login
```

[illegible]`stash@{1}`。

[illegible]

[illegible]

```
git stash show
```

[illegible]

```
git stash show stash@{n}
```

[illegible]

```
git stash show -p stash@{n}
```

[illegible]

[illegible]

```
git stash clear
```

□□□□□□□□□

```
git stash drop
```

□□□□□□□□□□□

```
git stash drop stash@{n}
```

□□□□□□□□

□□□□□□□□□□□□□□□□□□ - □□□

```
git stash pop
```

□□□□□□□□□□□□□□□□□ - □□□□

```
git stash pop stash@{n}
```

□□□□□□□□□

□□□□□□□□□□□□□□□□□□

```
git stash apply
```

□□□□□□□□□□□

```
git stash apply stash@{n}
```

□□□□□□□□□□□

□□□□git stash□□□□□□□□□□□□□

```
 git stash apply
```

□□□□□□□□□□□□□□

```
 git stash list
```

□□□□□□□□□□□□□

```
stash@{0}: WIP on master: 67a4e01 Merge tests into develop
stash@{1}: WIP on master: 70f0d95 Add user role to localStorage on user login
```

□□□□□□□git□□□□□□□□□□□□□□□□□□

```
git stash apply stash@{2}
```

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□。

```
git stash -p
```

[illegible]。

[illegible]2.13.0[illegible]**push**[illegible]pathspec[illegible]。

```
git stash push -m "My partial stash" -- app.config
```

[illegible]

[illegible]。

```
git checkout stash@{0} -- myfile.txt
```

[illegible]

Stashing[illegible] - [illegible] - [illegible]。

[illegible]

[illegible]

```
git stash --keep-index
```

[illegible]。

[illegible]

Stash[illegible]。 [illegible]

```
git stash -u
```

[illegible]。

[illegible]

[illegible]

```
git stash --patch
```

Git[illegible]。

[illegible]

[illegible]

```
git stash
git checkout correct-branch
git stash pop
```

[illegible] git stash pop[illegible]。 [illegible]

```
git stash apply
```

[illegible]

□□□□□□□□□□□□□□□□□□□□□□□□□□□□`git stash pop`□□□□□□□

```
$ git stash pop
[...]
Dropped refs/stash@{0} (2ca03e22256be97f9e40f08e6d6773c7d41dbfd1)
```

□□□`git stash drop`□□□□□□□□。□

□□□□□□□□□□□□□□□□

```
git fsck --no-reflog | awk '/dangling commit/ {print $3}'
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ - □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□□□□□`gitk` □

```
gitk --all $( git fsck --no-reflog | awk '/dangling commit/ {print $3}' )
```

□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□。

□□□□`git log --graph --oneline --decorate`□□`gitk` □□□□□□□□□□□□□GUI□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□

□*somebranch*□□ WIP□ *commithash*□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
git stash apply $stash_hash
```

□□□□□□□□gitk□□□□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□。

□□□□□□□□ https://riptutorial.com/zh-TW/git/topic/1440/□□

□□ 55: □□

□□

- `git show [options] <object> ...`

□□

□□□□Git□□。

- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□

Examples

□□

git show□□□□□Git□□。

□□□□□

□□□□□□□□□□□□□□□。

git show	
git show @~3	

□□□□□□□□

□□□□blob。

git show @~3:	3
git show @~3:src/program.js	src/program.js3blob
git show @:a.txt @:b.txt	b.txta.txt

□□□□□

□□□□□□□□□□□□。

□□□□□□□ https://riptutorial.com/zh-TW/git/topic/3030/□□

□□ 56: □□

□□

- git config [<file-option>] name [value] #git config□□□□□□□

□□

--system	Linux`$(prefix)/etc/gitconfig`
--global	Linux`~/.gitconfig`
--local	`.git/config`;

Examples

□□□□□□□□□□

□□□□Git□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。□shell□□□□

```
git config --global user.name "Mr. Bean"
git config --global user.email mrbean@example.com
```

- git config□□□□□□□□□□□
- --global□□□□□□□□□□□□□□□□□□□
- user.name□user.email□□□□□□□; user□□□□□□□□□。 name□email□□□□□□。
- "Mr. Bean"□mrbean@example.com□□□□□□□□□□□□。 □□□"Mr. Bean"□□□□□□□□□□□□□□□□□□□□□□□。

□□**git**□□

git□□□□□5□□□□

- 6□□□□
  - **%ALLUSERSPROFILE%\Git\Config** □□□Windows□
  - □□□□ **<git>/etc/gitconfig** □□□<git>□git□□□□。
    □□Windows□□□□**<git>\mingw64\etc\gitconfig** □
  - □□□□ **$XDG_CONFIG_HOME/git/config** □□□Linux / Mac□
  - □□□□ ~/.gitconfig □Windows□ %USERPROFILE%\.gitconfig □
  - □□□□ .git/config □□git repo $GIT_DIR □
  - □□□□□□ □□□git config -f □□□□□□□□□□□□□ git config -f .gitmodules ...
- □□**git -c**□□□□ □ git -c core.autocrlf=false fetch□□□□□□core.autocrlf□□□false □ □□□□fetch□□。

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

git config --system/global/local□□□□□3□□□□□□□□□git config -l□□□□□□ □□□□□。
“□□□”□□□□□□□□□□□□□□。

□git 2.8□□□□□□□□□□□□□□□□□□□□□□□□

```
git config --list --show-origin
```

□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□。

- □□core.editor□□□□。

  ```
  $ git config --global core.editor nano
  ```

- □□GIT_EDITOR□□□□。

  □□□□□□□

  ```
  $ GIT_EDITOR=nano git commit
  ```

  □□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□。

  ```
  $ export GIT_EDITOR=nano
  ```

- □□□□□□□□□□□□□□□□□□□□Git□□□□VISUAL□EDITOR□□□□。 □□VISUAL□EDITOR。□

  ```
  $ export EDITOR=nano
  ```

  □□□□□□□□□□□□□□□□□;□□shell□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□bash □□□□□□□□□~/.bashrc□~/.bash_profile 。□

□□□□□□□□□□□GUI□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□Git□□□Aborting commit due to empty commit message.□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□--wait□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□

□□

□□□□□□□□□□□□□□□□□□OS□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□**Windows**

□Microsoft Windows□□□□□OS□□□□□□□□□□□□□□□□ - □□+□□□CR + LF□。□□□□Unix□□□□Linux□OSX□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。□□□□Unix□□□□□□□□□□□□□LF□□□□□□□。

□□□□□□□□□□□□□□□□□□□□

```
git config --global core.autocrlf=true
```

□□□□ □□□□□□□□□Microsoft Windows□□□□□□□□□□□LF - > CR + LF□

□□Unix□Linux / OSX□

□□□□□Unix□□□□□□□□□□□□□□□Microsoft Windows□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□

```
git config --global core.autocrlf=input
```

□□□□ □□□□□CR + LF - > + LF□□□

□□□□□□□

□□□□□-c <name>=<value>□□□□□□□□□□□。

□□□□□□.gitconfig□□□□□□□□□□□□□□□□□□

```
git -c user.email = mail@example commit -m "some message"
```

□□□□□□□□□□□□□□□□□□user.name□user.email □git□□□□□□□□□□□□□。

□□□□

□□□□□□□□□□□□□□□git□

```
git config --global http.proxy http://my.proxy.com:portnumber
```

□□□□□□□□□□

```
git config --global --unset http.proxy
```

□□□□□□□

```
git config --global help.autocorrect 17
```

□□□□□git□□□□□□□□□□□□□□□□□□□□git stats□□□git status □。 □□□□help.autocorrect□□□□□□□□□□□□□ help.autocorrect□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□17□□□git□□□□□□autocorrected□□□□□□1.7□。

□□□□□□□□□□□□□□□□□□□□□git testingit□□□□□□□testingit is not a git command.

□□□□□□□□□

Git config□□□□□□git□□□□。 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□。

```
$ git config --list
...
core.editor=vim
credential.helper=osxkeychain
...
```

□□□□□□

```
$ git config <key> <value>
$ git config core.ignorecase true
```

□□□□□□□□□□□□□□□true□□□□--global

```
$ git config --global user.name "Your Name"
$ git config --global user.email "Your Email"
$ git config --global core.editor vi
```

□□□□□□□□□□□□。

□□□□□□□□□□□□

□Git 2.13□□□□□□□□□□□□□□□□□□□□□□□□□□□。

Windows□□□

□.gitconfig

□□□ git config --global -e

□□

```
[includeIf "gitdir:D:/work"]
  path = .gitconfig-work.config

[includeIf "gitdir:D:/opensource/"]
  path = .gitconfig-opensource.config
```

□□

- □□□□□□□□□□□□□“□□”。
- □□/□□ - □□"gitdir:D:/work"□□□□□。
- gitdir:□□□□□□。

□.gitconfig-work.config

□□□*.gitconfig*□□□□□□□

```
[user]
  name = Money
  email = work@somewhere.com
```

□.gitconfig-opensource.config

□□□*.gitconfig*□□□□□□□

```
[user]
  name = Nice
  email = cool@opensource.stuff
```

Linux□□□

```
[includeIf "gitdir:~/work/"]
  path = .gitconfig-work
[includeIf "gitdir:~/opensource/"]
  path = .gitconfig-opensource
```

Windows□□□□□□□□□。

□□□□□□□ https://riptutorial.com/zh-TW/git/topic/397/□□

□□ 57: □□□□□□

Examples

□□□□□

□□□git merge□□□□□□git□□□“□□□□”□□。 □□□□□□□□□□□□□□□□□□□□□□。

□□□□□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/or "git commit -a")
```

Git□□□□□□□□□□□□□□□□□□□□

```
<<<<<<<<< HEAD: index.html #indicates the state of your current branch
<div id="footer">contact : email@somedomain.com</div>
========= #indicates break between conflicts
<div id="footer">
please contact us at email@somedomain.com
</div>
>>>>>>>>> iss2: index.html #indicates the state of the other branch (iss2)
```

□□□□□□□□□□□□□□□<<<<<<□>>>>>>>□□□□□□□□□□□□□□<<<<<<<, >>>>> >>□□========□□。 □□git add index.html□□□□□□□ git commit□□□□。

□□□□□□□□□□□ https://riptutorial.com/zh-TW/git/topic/3233/□□□□□□

□□ 58: □□□Git

Examples

□□Atlassian□□□□□□□SVN□□□Git

□□□□□Atlassian□□□□□□。□□□□□□Java□□□□□□□□□□□□□□□□□□□□□□Java Runtime Environment JRE。

□□□□java -jar svn-migration-scripts.jar verify□□□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□Git□subversion□git-svn□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□Git□□□□□□□□□。

□□□□□□□□□□□□authors□□。 Subversion□□□□□□□□□□□□□□□□□□。□□□Git□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□subversion□□□□□□□□□□Git□□□□

```
java -jar svn-migration-scripts.jar authors <svn-repo> authors.txt
```

□□<svn-repo>□□□□□□□subversion□□□□URL。□□□□□□□□□□□□□□□□□□□□□□□authors.txt。□□□□□□□□□□□□<username>@mycompany.com。□authors□□。□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□svn repo□□□Git□

```
git svn clone --stdlayout --authors-file=authors.txt <svn-repo> <git-repo-name>
```

□□<svn-repo>□□□□□□□□□□URL□□□ <git-repo-name>□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□

- □□□--stdlayout□□□□Git□□□□□□□□trunk □ branches□tags□□□□□□□□□□。□□□□□□□□Subversion□□□□□□□□□□trunk□□□□any / all branch□□□□tags□□□□□□□。□□□□□□□□□□□□□□□□ git svn clone --trunk=/trunk --branches=/branches --branches=/bugfixes --tags=/tags --authors-file=authors.txt <svn-repo> <git-repo-name>。
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。
- □□□□□□□□□□□□□□□□□□□□□□□□subversion□□□□□□□□□□□□□□□□□□□□□□□□□□□。

git svn clone□subversion□□□□trunk□□□□□□□□□□□□□□□□□subversion□□□□tags/□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Linux□□□□□□□□□□□□□。□□□□□□□□ git branch -a□□□□□□□□□□□□□□ git tag -l□□□□□□□□□□。

```
git for-each-ref refs/remotes/origin/tags | cut -d / -f 5- | grep -v @ | while read tagname;
do git tag $tagname origin/tags/$tagname; git branch -r -d origin/tags/$tagname; done
git for-each-ref refs/remotes | cut -d / -f 4- | grep -v @ | while read branchname; do git
branch "$branchname" "refs/remotes/origin/$branchname"; git branch -r -d "origin/$branchname";
done
```

□svn□Git□□□□□□□□□□□□□push□□□□□□□push□□□□□□□□□□□□□□□□□□Git□□□□□□□□svn□□□□□□□□□□□□□□□□□。

SubGit

SubGit□□□□□□□□□□□□SVN□□□□git。

```
$ subgit import --non-interactive --svn-url http://svn.my.co/repos/myproject myproject.git
```

□□svn2git□SVN□□□Git

svn2git□□□Ruby□□□□□□□□git-svn□□git□□□SVN□□□□□□□□□□□□Subversion□□□Git□□□□□□□□□□□□trunk□tags□branches□□□□□。

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□svn□□□□□□□□□□□□

```
$ svn2git http://svn.example.com/path/to/repo
```

□□□□□□□□□□□svn□□□□

```
$ svn2git http://svn.example.com/path/to/repo --trunk trunk-dir --tags tags-dir --branches
branches-dir
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□`--notrunk` □`--` `--nobranches`□`--notags`。

□□□ `$ svn2git http://svn.example.com/path/to/repo --trunk trunk-dir --notags --nobranches`□□□□□□□□□□。

□□□

```
$ svn2git http://svn.example.com/path/to/repo --exclude build --exclude '.*\.zip$'
```

□□□□□

□□□□□□□□git□□。 □□□□□□□□□□□□

```
$ git gc --aggressive
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□。

□Team Foundation□□□□□TFVC□□□□Git

□□□□□□□Git-TF□□□□□□□□□□□□□□□□□git。 □□□□□□□tfs checkins□□□git□□□□□□□□□□□□□。

□□□Git-TF□□□□□□□□□Git□□□□□□□□□□□

□□Git-TF

□□□□Codeplex□□□□□□□Git-TF□ Git-TF @ Codeplex

□□□□**TFVC**□□□□

□□powershell□win□□□□□□

```
git-tf clone http://my.tfs.server.address:port/tfs/mycollection
'$/myproject/mybranch/mysolution' --deep
```

--deep□□□□□□□□□□□□□□□□Git-Tf□□□□□□□□□□。 □□□□□□□□cloe□□□□□□□□□□□□git□□□。

□□

- □□.gitignore□□。 □□□□□□□Visual Studio□□□□□□□□□□□□□□□□□□□□□□github / gitignore□□□□□□□□□□□□□□
  。
- □□□□□□□□TFS□□□□□□□□□□□□* .vssscc□□□。 □□□□□□□□GlobalSection□TeamFoundationVersionControl□□□□
  □□□□□□□□...... EndClobalSection

□□□□□

□□□□□□□□□□□□□□□□□□□□□。

```
git add .
git commit -a -m "Coverted solution source control from TFVC to Git"

git remote add origin https://my.remote/project/repo.git
git push origin master
```

□Mercurial□□□Git

□□□□□□□□□Mercurial Repo□□Git □

1. □□□□□□ □

```
cd
git clone git://repo.or.cz/fast-export.git
git init git_repo
cd git_repo
~/fast-export/hg-fast-export.sh -r /path/to/old/mercurial_repo
git checkout HEAD
```

2. □□Hg-Git □□□□□□□□□□□□□□□ https □ //stackoverflow.com/a/31827990/5283213

3. □□GitHub□□□□ □□□□□□□□□□GitHub□ 。

□□□□□□□Git□ https://riptutorial.com/zh-TW/git/topic/3026/□□□git

□□ 59: □□□

□□

- `git mv <source> <destination>`
  - `git mv -f <source> <destination>`

□□

-f--force	

Examples

□□□□□□

□□□□□oldName□□□□newName

```
git mv directoryToFolder/oldName directoryToFolder/newName
```

□□□□git commit□/□git push

□□□□□□□□

> □□□□□□'directoryToFolder / oldName'□□□□□□□

□□□□□□□

```
git mv directoryToFolder/oldName temp && git mv temp directoryToFolder/newName
```

□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□

```
git branch -m old_name new_name
```

□□□□□□□□□□

□□□□□□□□□□□□□□

```
git checkout old_branch
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
git branch -m new_branch
git push origin :old_branch
git push --set-upstream origin new_branch
```

□□□□□□□□ https://riptutorial.com/zh-TW/git/topic/1814/□□□

□□ 60: □

□□

- □.git /□/ applypatch-MSG
- □.git /□/□□-MSG
- git□/□/□□□□
- □.git /□/□applypatch
- □.git /□/□□□
- □.git /□/□□□□-MSG
- □.git /□/□□
- □.git /□/□□□
- □.git /□/□□

□□

--no-verify□-n□□□□git□□□□□□□□□。
□□□ git commit -n

□□□□□□□□□□□Git□□□Atlassian。

Examples

□□-MSG

□□□□□□prepare-commit-msg□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□□prepare-commit-msg □□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□□

```
word="ticket [0-9]"
isPresent=$(grep -Eoh "$word" $1)

if [[ -z $isPresent ]]
  then echo "Commit message KO, $word is missing"; exit 1;
  else echo "Commit message OK"; exit 0;
fi
```

□□□□

□□□□□□□□□□□□□□□□□。□□。□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□pre-commit□prepare-commit-msg□commit-msg□post-commit□post-checkout□pre-rebase。

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。□□□□□□□□git checkout□git rebase□□□□□□□□□□□□□□□□。

□□“pre-”□□□□□□□□□□□□□□□□□□“post-”□□□□□□□□。

□□□

□□□□□□□□□□□□post-commit□□□□□□□□git checkout□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□

1. □□HEAD□□□□

2. □HEAD□□□□□
3. □□□□□□□□□□□□□□□□□□□□□□□□1□0 □。

□□□□□□□git checkout□□□□□□。

□□□

□commit-msg□□□□□□□□□□□。 □□□□□git commit□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□

□□□□□□□□□□□□。 □□□□□□□□□。

□□□□□□□□□□□□□□□□□□□pre-receive□□□□□□

```
<old-value> <new-value> <ref-name>
```

□□□

□□□□git commit□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□。

□□□。 □□□□□□□□□□□□□□□EOL□□。

□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□-MSG

□pre-commit□□□□□□□□□□□□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□

- □□□□□□□□□□□□。
- □□□□□
  - □□□ -m□-F□□□□
  - □□□ -t□□□□
  - □□□□□□□□□□□□□
  - □□□□□□□□□□□□□□□。
- □□□□□SHA1□□。 □□□□-c □ -C□--amend□□□□□□□□□□。

□pre-commit□□□□□□□□□□□□□□□□。

□□□

□git rebase□□□□□□□□□□□□□□□□。 □□□□□□□□□rebase□□□□□□。

□□□□□2□□□□

1. □□□□□□□□□□□
2. □□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□rebase□□。

□□□

□□□□□□git push□□□git push□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□。

□□□□□□□□□□□□□□□□ref□□□□□□□□□□□□□□□□□□□□□

```
<old-value> <new-value> <ref-name>
```

□□

□pre-receive□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

□□□□□□□□3□□□□

- □□□□ref□□□□
- □□□ref□□□□□□□□
- □□□ref□□□□□□□。

□□□□□pre-receive□□□□□□□□□□□ref□□□□□□update □□□□□□□□□□□□□□□□□□□□□□。

□□

□□□*Git 1.8.2*□□□□□。

1.8

□□□□□□□□□□□□。□□□。

□□□.git/hooks/□□ **gotcha alert** □□□□□□pre-push□□□□□□□pre-push .git/hooks/ □□□□□□□□□ chmod +x ./git/hooks/pre-push。

□□□□□·□□□ □ Hannah Wolfe□□□□□□□□□□□□□

```
#!/bin/bash

protected_branch='master'
current_branch=$(git symbolic-ref HEAD | sed -e 's,.*/\(.*\),\1,')

if [ $protected_branch = $current_branch ]
then
    read -p "You're about to push master, is that what you intended? [y|n] " -n 1 -r <
/dev/tty
    echo
    if echo $REPLY | grep -E '^[Yy]$' > /dev/null
    then
        exit 0 # push will execute
    fi
    exit 1 # push will not execute
else
    exit 0 # push will execute
fi
```

□□Volkan Unsal□□□□□□□□□RSpec□□□□□□□□□□□□

```
#!/usr/bin/env ruby
require 'pty'
html_path = "rspec_results.html"
begin
  PTY.spawn( "rspec spec --format h > rspec_results.html" ) do |stdin, stdout, pid|
  begin
    stdin.each { |line| print line }
  rescue Errno::EIO
  end
end
rescue PTY::ChildExited
  puts "Child process exit!"
end

# find out if there were any errors
html = open(html_path).read
examples = html.match(/(\d+) examples/)[0].to_i rescue 0
errors = html.match(/(\d+) errors/)[0].to_i rescue 0
if errors == 0 then
  errors = html.match(/(\d+) failure/)[0].to_i rescue 0
end
pending = html.match(/(\d+) pending/)[0].to_i rescue 0

if errors.zero?
  puts "0 failed! #{examples} run, #{pending} pending"
  # HTML Output when tests ran successfully:
  # puts "View spec results at #{File.expand_path(html_path)}"
  sleep 1
  exit 0
else
  puts "\aCOMMIT FAILED!!"
  puts "View your rspec results at #{File.expand_path(html_path)}"
  puts
  puts "#{errors} failed! #{examples} run, #{pending} pending"
  # Open HTML Ooutput when tests failed
  # `open #{html_path}`
  exit 1
end
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□exit 0□□□□□□□exit 1。□□□□exit 1 □□□□□□□□□□□□□□□□□□□git push...□□□□□ git push...

□□□□□□□□□□□□□□□□□□□□□□□□□□□“--no-verify”□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□。

□□□ https □ //git-scm.com/docs/githooks#_pre_push
□□□□□ https □ //github.com/git/git/blob/87c86dd14abe8db7d00b0df5661ef8cf147a72a3/templates/hooks--pre-push.sample

□□□□□□□**Maven**□□□□□□□□□□□

.git/hooks/pre-commit

```
#!/bin/sh
if [ -s pom.xml ]; then
    echo "Running mvn verify"
    mvn clean verify
    if [ $? -ne 0 ]; then
        echo "Maven build failed"
        exit 1
    fi
fi
```

□□□□□□□□□□□□□□□

post-receive hooks□□□□□□□□□□□□□□□□□□□。

```
$ cat .git/hooks/post-receive

#!/bin/bash

IFS=' '
while read local_ref local_sha remote_ref remote_sha
do

  echo "$remote_ref" | egrep '^refs\/heads\/[A-Z]+-[0-9]+$' >/dev/null && {
    ref=`echo $remote_ref | sed -e 's/^refs\/heads\///'`
    echo Forwarding feature branch to other repository: $ref
    git push -q --force other_repos $ref
  }

done
```

□□□□□□ egrep regexp□□□□□□□□□□□□□JIRA-12345□□□□Jira□□□。 □□□□□□□□□□□□□□□□□□□□□。

□□□□□□ https://riptutorial.com/zh-TW/git/topic/1330/□

□□ 61: □□□

□□

- `git rev-list [options] <commit> ...`

□□

-	。

Examples

List□master□□□□□□origin / master□□□

```
git rev-list --oneline master ^origin/master
```

Git rev-list□□□□□□□□□□□□□□□□□□□□。□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

- □□--oneline□□□□□□□□□□□□。
- ^□□□□□□□□□□□□□□□□□。
- □□□□□□□□□□□□□□□□□。□□□ git rev-list foo bar ^baz□□foo□bar□□□□□□□□□baz。

□□□□□□□□ https://riptutorial.com/zh-TW/git/topic/431/□□□

□□

S. No	□	Contributors
1	□□□□Git	Ajedi32, Ala Eddine JEBALI, Allan Burleson, Amitay Stern, Andy Hayden, AnimiVulpis, ArtOfWarfare, bahrep, Boggin, Brian, Community, Craig Brett, Dan Hulme, ericdwang, eykanal, Fernando Hoces De La Guardia, Fred Barclay, Henrique Barcelos, intboolstring, Irfan, Jackson Blankenship, janos, Jav_Rock, jeffdill2, JonasCz, JonyD, Joseph Dasenbrock, Kageetai, Karthik , KartikKannapur, Kayvan N, Knu, Lambda Ninja, maccard, Marek Skiba, Mateusz Piotrowski, Mingle Li, mouche, Nathan Arthur, Neui, NRKirby, ob1, ownsourcing dev training, Pod, Prince J, RamenChef, Rick, Roald Nefs, ronnyfm, Sazzad Hissain Khan, Scott Weldon, Sibi Raj, TheDarkKnight, theheadofabroom, ɟolɐəz əɥʇ qoq, Tot Zam, Tyler Zika, tymspy, Undo, VonC
2	.mailmap□□□□□□□□□□□□□□□	Mario, Michael Plotke
3	Bash Ubuntu□□Git□□□□	Manishh
4	DIFF□	fybw id
5	Git Clean	gnis, MayeulC, n0shadow, pktangyue, Priyanshu Shekhar, Ralf Rafael Frix
6	Git Diff	Aaron Critchley, Abhijeet Kasurde, Adi Lester, anderas, apidae, Brett, Charlie Egan, eush77, □□□□□□, intboolstring, Jack Ryan, JakeD, Jakub Narębski, jeffdill2, Joseph K. Strauss, khanmizan, Luke Taylor, Majid, mnoronha, Nathaniel Ford, Ogre Psalm33, orkoden, Ortomala Lokni, penguincoder, pylang, SurDin, Will, ydaetskcoR, Zaz
7	Git GUI□□□	Alu, Daniel Käfer, Greg Bray, Nemanja Trifunovic, Pedro Pinheiro
8	Git Patch	Dartmouth, Liju Thomas
9	Git Remote	AER, ambes, Dániel Kis, Dartmouth, Elizabeth, Jav_Rock, Kalpit, RamenChef, sonali, sunkuet02
10	Git rerere	Isak Combrinck
11	git send-email	Aaron Skomra, Dong Thang, fybw id, Jav_Rock, kofemann
12	GIT-SVN	Bryan, Randy, Ricardo Amores, RobPethi
13	GIT-TFS	Boggin, Kissaki
14	Git□□□□□	Ates Goral
15	Git□□□□	Dartmouth, Jakub Narębski
16	Git□□□□□□LFS□	Alex Stuckey, Matthew Hallatt, shoelzer
17	Git□□□□□	Kelum Senanayake, kiamlaluno
18	Git□□	Atul Khanduri, demonplus, TheDarkKnight

19	Git□□	Dartmouth, Farhad Faghihi, Hugo Buff, KartikKannapur, lxer, penguincoder, RamenChef, SashaZd, Tyler Hyndman, vkluge
20	Reflog - □□git□□□□□□□□□	Braiam, Peter Amidon, Scott Weldon
21	TortoiseGit	Julian, Matas Vaitkevicius
22	Worktrees	andipla, Confiqure, Victor Schröder
23	□□.gitattributes□□	Chin Huang, dahlbyk, Toby
24	□□filter-branch□□□□□□	gavinbeatty, gavv, Glenn Smith
25	□□Gitk□□□□□□□□□□□□□□	orkoden
26	□□□□□	Boggin, Caleb Brinkman, forevergenin, heitortsergent, intboolstring, jeffdill2, Julie David, Kalpit, Matt Clark, MByD, mnoronha, mpromonet, mystarrocks, Pascalz, Raghav, Ralf Rafael Frix, Salah Eddine Lahniche, Sam, Scott Weldon, Stony, Thamilan, Vivin George, VonC, Zaz
27	□□□□□	AER, Andrea Romagnoli, Andy Hayden, Blundering Philosopher, Dartmouth, Ezra Free, ganesshkumar, □□□□□, kartik, KartikKannapur, mnoronha, Peter Mitrano, pkowalczyk, Rick, Undo, Wojciech Kazior
28	□□	nighthawk454
29	□□	AesSedai101, Andy Hayden, Asaph, Confiqure, intboolstring, Jakub Narębski, jkdev, Muhammad Abdullah, Nathan Arthur, ownsourcing dev training, Richard Dally, Wolfgang
30	□□□□□□□□	Boggin, Confiqure, Daniel Käfer, Dimitrios Mistriotis, forresthopkinsa, hardmooth, Horen, Kissaki, Majid, Sardathrion, Scott Weldon
31	□□	Amitay Stern, Andrew Kay, AnimiVulpis, Bad, BobTuckerman, Community, dan, Daniel Käfer, Daniel Stradowski, Deepak Bansal, djb, Don Kirkby, Duncan X Simpson, Eric Bouchut, forevergenin, fracz, Franck Dernoncourt, Fred Barclay, Frodon, gavv, Irfan, james large, janos, Jason, Joel Cornett, Jon Schneider, Jonathan, Joseph Dasenbrock, jrf, kartik, KartikKannapur, khanmizan, kirrmann, kisanme, Majid, Martin, MayeulC, Michael Richardson, Mihai, Mitch Talmadge, mkasberg, nepda, Noah, Noushad PP, Nowhere man, olegtaranenko, Ortomala Lokni, Ozair Kafray, PaladiN, □ANAYI□TIS, Priyanshu Shekhar, Ralf Rafael Frix, Richard Hamilton, Robin, RudolphEst, Siavas, Simone Carletti, the12, Uwe, Vlad, wintersolider, Wojciech Kazior, Wolfgang, Yerko Palma, Yury Fedorov, zygimantus
32	□□	AesSedai101, Ajedi32, Andy, Anthony Staunton, Asenar, bstpierre, erewok, eush77, fracz, Gaelan, jrf, jtbandes, madhead, Michael Deardeuff, mickeyandkaka, nus, penguincoder, riyadhalnur, thanksd, Tom Hale, Wojciech Kazior, zinking
33	□□	brentonstrine, Liam Ferris, Noah, penguincoder, Undo, Vogel612, Wolfgang
34	□□□□□□□□□□	Keyur Ramoliya, RamenChef
35	□□	AER, Alexander Bird, anderas, Ashwin Ramaswami, Braiam, BusyAnt

		, Confiqure, Daniel Käfer, Derek Liu, Dunno, e.doroskevic, Enrico Campidoglio, eskwayrd, □□□□□, Hugo Ferreira, intboolstring, Jeffrey Lin, Joel Cornett, Joseph K. Strauss, jtbandes, Julian, Kissaki, LeGEC, Libin Varghese, Luca Putzu, lucash, madhukar93, Majid, Matt, Matthew Hallatt, Menasheh, Michael Mrozek, Nemanja Boric, Ortomala Lokni, Peter Mitrano, pylang, Richard, takteek, Travis, Victor Schröder, VonC, Wasabi Fan, yarons, Zaz
36	□□□□□□difftools	AesSedai101, Micha Wiedenmann
37	□□□	321hendrik, Chin Huang, ComicSansMS, foraidt, intboolstring, J F, kowsky, mpromonet, PaladiN, tinlyx, Undo, VonC
38	□□	4444, Jeff Puckett
39	□□/□□□□□□	4444, Hannoun Yassir, jornh, Kissaki, MrTux, Scott Weldon, Simone Carletti, zebediah49
40	□□□□□□□□□	AER, AesSedai101, agilob, Alex, Amitay Stern, AnimiVulpis, Ates Goral, Aukhan, Avamander, Ben, bpoiss, Braiam, bwegs, Cache Staheli, Collin M, Community, Dartmouth, David Grayson, Devesh Saini, Dheeraj vats, eckes, Ed Cottrell, enrico.bacis, Everettss, Fabio, fracz, Franck Dernoncourt, Fred Barclay, Functino, geek1011, Guillaume Pascal, HerrSerker, intboolstring, Irfan, Jakub Narębski, Jeff Puckett, Jens, joaquinlpereyra, John Slegers, JonasCz, Jörn Hees, joshng, Kačer, Kapep, Kissaki, knut, LeftRight92, Mackattack, Marvin, Matt, MayeulC, Mitch Talmadge, Narayan Acharya, Nathan Arthur, Neui, noɥʇʎdʎzɐɹƆ, Nuri Tasdemir, Ortomala Lokni, PaladiN, Panda, peci1, pktangyue, poke, pylang, RhysO, Rick, rokonoid, Sascha, Scott Weldon, Sebastianb, SeeuD1, sjas, Slayther, SnoringFrog, spikeheap, theJollySin, Toby, ʇolɐəz əɥʇ qoq, Tom Gijselinck, Tomasz Bąk, Vi., Victor Schröder, VonC, Wilfred Hughes, Wolfgang, ydaetskcoR, Yosvel Quintero, Yury Fedorov, Zaz, Zeeker
41	□□	Creative John, Hardik Kanjariya ツ, Julie David, kisanme, □ANAYI□TIS, Scott Weldon, strangeqargo, Zaz
42	□	Kissaki, MayeulC, mpromonet, rene, Ryan, Scott Weldon, Shog9, Stony, Thamilan, Thomas Gerot, Zaz
43	□□	jwd630
44	□□□□	Atul Khanduri, Braiam, bud-e, dubek, Florian Hämmerle, intboolstring, Julian, kisanme, Lochlan, mpromonet, RedGreenCode
45	□□	AER, Cody Guldner, cringe, frlan, Guillaume, intboolstring, Mário Meyrelles, Marvin, Matt S, MayeulC, pcm, pogosama, Thomas Gerot, Tomás Cañibano
46	□□	Aaron Critchley, AER, Alan, Allan Burleson, Amitay Stern, Andrew Sklyarevsky, Andy Hayden, Anonymous Entity, APerson, bandi, Cache Staheli, Chris Forrence, Cody Guldner, cormacrelf, davidcondrey, Deep, depperm, ericdwang, Ethunxxx, Fred Barclay, George Brighton, Igor Ivancha, intboolstring, JacobLeach, James Taylor, janos, joeytwiddle, Jordan Knott, KartikKannapur, kisanme, Majid, Matt Clark, Matthew Hallatt, MayeulC, Micah Smith, Pod, Rick, Scott Weldon, SommerEngineering, Sonny Kim, Thomas Gerot, Undo, user1990366, vguzmanp, Vladimir F, Zaz

47	□□	Adi Lester, AesSedai101, Alexander Bird, Andy Hayden, Boggin, brentonstrine, Brian, Colin D Bennett, ericdwang, Karan Desai, Matthew Hallatt, Nathan Arthur, Nathaniel Ford, Nithin K Anil, Pace, Rick, textshell, Undo, Zaz
48	□□	adarsh, ams, AndiDog, bandi, Braiam, Caleb Brinkman, eush77, georgebrock, jpkrohling, Julian, Mateusz Piotrowski, Ortomala Lokni, RamenChef, Tall Sam, WMios
49	□□□□□□□□□□□□	Thomas Crowley
50	□□git□□□□□	xiaoyaoworm
51	□□	Dartmouth, forevergenin, Neto Buenrostro, RamenChef
52	□□	fracz, Matthew Hallatt, nighthawk454, Priyanshu Shekhar, WPrecht
53	□□□□□□	Ahmed Metwally, Andy Hayden, Aratz, Atif Hussain, Boggin, Brett, Confiqure, davidcondrey, Fabio, Flows, fracz, Fred Barclay, guleria, intboolstring, janos, jaredr, Kamiccolo, KraigH, LeGEC, manasouza, Matt Clark, Matthew Hallatt, MByD, mpromonet, Muhammad Abdullah, Noah, Oleander, Pedro Pinheiro, RedGreenCode, Toby Allen, Vogel612, ydaetskcoR
54	□□	aavrug, AesSedai101, Asaph, Brian Hinchey, bud-e, Cache Staheli, Deep, e.doroskevic, fracz, GingerPlusPlus, Guillaume, inkista, Jakub Narębski, Jarede, jeffdill2, joeytwiddle, Julie David, Kara, Koraktor, Majid, manasouza, Ortomala Lokni, Patrick, Peter Mitrano, Ralf Rafael Frix, Sebastianb, Tomás Cañibano, Wojciech Kazior
55	□□	Zaz
56	□□	APerson, Asenar, Cache Staheli, Chris Rasys, e.doroskevic, Julian, Liyan Chang, Majid, Micah Smith, Ortomala Lokni, Peter Mitrano, Priyanshu Shekhar, Scott Weldon, VonC, Wolfgang
57	□□□□□□	Braiam, Dartmouth, David Ben Knoble, Fabio, nus, Vivin George, Yury Fedorov
58	□□□Git	AesSedai101, Boggin, Confiqure, Guillaume Pascal, Indregaard, Rick, TheDarkKnight
59	□□□	bud-e, Karan Desai, P.J.Meisch, PhotometricStereo
60	□	AesSedai101, AnoE, Christiaan Maks, Confiqure, Eidolon, Flows, fracz, kaartic, lostphilosopher, mwarsco
61	□□□	mkasberg