



免費電子書

學習

# Git

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#git

.....	1
<b>1: Git</b> .....	<b>2</b>
.....	2
.....	2
Examples.....	3
.....	3
.....	4
.....	5
.....	5
.....	6
.....	6
GitSSH.....	7
Git.....	7
<b>2: .mailmap</b> .....	<b>10</b>
.....	10
.....	10
Examples.....	10
.....	10
<b>3: Bash UbuntuGit</b> .....	<b>11</b>
.....	11
Examples.....	11
.....	11
<b>4: DIFF</b> .....	<b>12</b>
.....	12
Examples.....	12
.....	12
.....	12
.....	12
<b>5: Git Clean</b> .....	<b>14</b>
.....	14
.....	14

Examples.....	14
.....	14
.....	14
.....	14
.....	14
<b>6: Git Diff.....</b>	<b>15</b>
.....	15
.....	15
Examples.....	15
.....	15
.....	15
.....	16
.....	16
meld.....	16
.....	16
word-diff.....	17
.....	17
.....	17
Diff UTF-16plist.....	17
.....	18
.....	18
.....	18
.....	18
<b>7: Git GUI.....</b>	<b>20</b>
Examples.....	20
GitHub.....	20
Git Kraken.....	20
SourceTree.....	20
gitkgit-gui.....	20
SmartGit.....	22
Git.....	22
<b>8: Git Patch.....</b>	<b>24</b>
.....	

.....	24
Examples.....	25
.....	25
.....	25
<b>9: Git Remote.....</b>	<b>26</b>
.....	26
.....	26
Examples.....	27
.....	27
.....	27
.....	27
.....	27
GitURL.....	27
.....	28
<b>10: Git rerere.....</b>	<b>29</b>
.....	29
Examples.....	29
rerere.....	29
<b>11: git send-email.....</b>	<b>30</b>
.....	30
.....	30
Examples.....	30
Gmailgit send-email.....	30
.....	30
.....	31
<b>12: GIT-SVN.....</b>	<b>32</b>
.....	32
.....	32
Examples.....	32
SVN.....	32
SVN.....	33

SVN.....	33
.....	33
.....	33
<b>13: GIT-TFS.....</b>	<b>35</b>
.....	35
Examples.....	35
git-tfs.....	35
gitgit-tfs.....	35
git-tfsChocolatey.....	35
git-tfs.....	35
git-tfs.....	36
<b>14: Git.....</b>	<b>37</b>
Examples.....	37
Git.....	37
<b>15: Git.....</b>	<b>38</b>
.....	38
Examples.....	38
.....	38
.....	38
HEAD.....	38
Reflog @ { }.....	38
Reflog @ { }.....	39
/ @{}.....	39
^ .....	39
^ 0 ^ { }.....	40
^ {/ } / .....	40
<b>16: GitLFS.....</b>	<b>41</b>
.....	41
Examples.....	41
LFS.....	41
.....	41
LFS.....	41

<b>17: Git</b>	<b>42</b>
.....	42
Examples	42
.....	42
Git	42
<b>18: Git</b>	<b>43</b>
.....	43
.....	43
Examples	43
.....	43
GIT	43
<b>19: Git</b>	<b>45</b>
.....	45
.....	45
Examples	45
.....	45
.....	45
.....	45
.....	45
.....	46
.....	46
Git	46
.....	46
<b>20: Reflog - git</b>	<b>47</b>
.....	47
Examples	47
rebase	47
<b>21: TortoiseGit</b>	<b>48</b>
Examples	48
.....	48
.....	48
.....	50

.....	50
.....	51
.....	51
.....	52
22: Worktrees.....	54
.....	54
.....	54
.....	54
Examples.....	54
.....	54
.....	54
23: .gitattributes.....	56
Examples.....	56
.....	56
.....	56
.....	56
.gitattribute.....	56
24: filter-branch.....	57
Examples.....	57
.....	57
git committercommit author.....	57
25: Gitk.....	58
Examples.....	58
.....	58
.....	58
.....	58
26: .....	59
.....	59
Examples.....	59
.....	59
.....	59

LS-.....	59
.....	60
.....	60
.....	60
.....	60
.....	60
.....	60
.....	60
.....	60
.....	61
GitURL.....	61
.....	61
URL.....	62
URL.....	62
<b>27: .....</b>	<b>63</b>
.....	63
Examples.....	63
.....	63
.....	63
.....	63
.....	64
.....	64
<b>28: .....</b>	<b>65</b>
Examples.....	65
.....	65
.....	65
HEAD.....	65
.....	65
.....	66
.....	66
.....	66
.....	66



Blob.....	67
.....	67
HEAD.....	67
.....	67
.....	67
<b>29: .....</b>	<b>69</b>
.....	69
Examples.....	69
.....	69
.....	69
.....	69
.....	69
.....	69
hunk.....	70
.....	71
<b>30: .....</b>	<b>72</b>
.....	72
Examples.....	72
Gitflow.....	72
.....	74
.....	74
.....	74
GitHub.....	74
<b>31: .....</b>	<b>76</b>
.....	76
.....	76
.....	76
Examples.....	76
.....	76
.....	77
.....	77
.....	78

.....	78
.....	78
.....	78
.....	79
.....	79
HEAD.....	79
.....	79
.....	79
<b>32: .....</b>	<b>80</b>
Examples.....	80
.....	80
/.....	80
.....	80
.....	80
.....	81
.....	81
.....	82
.gitignore.....	82
Unstage.....	83
<b>33: .....</b>	<b>84</b>
.....	84
.....	84
Examples.....	84
.....	84
.....	84
.....	84
.....	84
.....	85
.....	85
<b>34: .....</b>	<b>86</b>
Examples.....	86
.....	86
.....	

.....	86
.....	86
<b>35: .....</b>	<b>87</b>
.....	87
.....	87
.....	87
Examples .....	87
.....	87
Rebase.....	88
.....	<b>88</b>
.....	88
.....	88
Rebase.....	88
.....	<b>89</b>
.....	<b>89</b>
.....	<b>89</b>
.....	<b>89</b>
Rebase.....	89
.....	90
.....	90
.....	90
.....	<b>90</b>
.....	<b>90</b>
.....	<b>90</b>
.....	<b>90</b>
.....	<b>92</b>
git-pullrebase.....	92
rebase.....	92
.....	92

<b>36: difftools</b>	<b>93</b>
Examples	93
	93
KDiff3	93
KDiff3diff	93
IntelliJ IDEWindows	93
IntelliJ IDEdiffWindows	94
<b>37:</b>	<b>95</b>
Examples	95
	95
Git	95
	95
	95
	96
	96
<b>38:</b>	<b>98</b>
	98
	98
Examples	98
	98
	<b>98</b>
	<b>98</b>
<b>Backport</b>	<b>98</b>
<b>39: /</b>	<b>100</b>
	100
Examples	100
git bisect	100
	101
<b>40:</b>	<b>102</b>
	102
Examples	102

.gitignore.....	102
.....	102
.gitignore.....	103
.....	104
.gitignore.....	105
.gitignore.....	106
Git.....	106
.....	106
gitignore.....	107
.....	107
.....	108
.gitignore.....	108
.....	109
[stub].....	109
[].....	110
.gitignore.....	111
.....	111
.gitignore.....	111
<b>41: .....</b>	<b>113</b>
Examples.....	113
.....	113
.....	113
.....	113
.....	113
.....	113
Git.....	113
git stash.....	114
<b>42: .....</b>	<b>116</b>
.....	116
.....	116
.....	116
.....	

Examples.....	116
.....	116
.....	117
.....	117
.....	117
.....	117
.....	117
.....	117
.....	117
.....	118
.....	118
.....	118
.....	118
.....	118
.....	119
.....	119
Examples.....	119
git.....	119
.....	120
.....	120
.....	120
.....	120
Examples.....	120
.....	120
.....	121
.....	121
.....	121
.....	122
.....	122
.....	122
.....	122

[illegible]

.....	128
.....	129
.....	129
.....	129
.....	129
git.....	130
.....	130
.....	130
.....	131
.....	131
GPG.....	131
47: .....	133
Examples.....	133
.....	133
reflog.....	133
.....	134
.....	134
.....	135
/.....	135
48: .....	137
.....	137
.....	137
.....	137
Examples.....	137
.....	137
Rebase.....	137
Autosquashrebase.....	138
.....	139
.....	139
49: .....	140



Examples.....	140
.....	140
<b>50: git.....</b>	<b>141</b>
.....	141
Examples.....	141
.....	141
<b>51: .....</b>	<b>142</b>
.....	142
.....	142
Examples.....	142
git.....	142
git.....	142
git.....	143
<b>52: .....</b>	<b>144</b>
.....	144
.....	144
.....	144
Examples.....	144
.....	144
.....	145
.....	145
.....	145
<b>53: .....</b>	<b>146</b>
.....	146
.....	146
.....	146
Examples.....	146
“”Git.....	146
Online.....	146
.....	147
.....	147
.....	147

.....	148
.....	148
.....	149
.....	149
.....	150
Git.....	150
.....	150
.....	150
git log.....	150
<b>54: .....</b>	<b>152</b>
.....	152
.....	152
.....	152
Examples.....	152
Stashing.....	152
.....	154
.....	154
.....	154
.....	154
.....	155
.....	155
.....	155
.....	155
.....	156
.....	156
.....	156
.....	157
<b>55: .....</b>	<b>158</b>
.....	158
.....	158
Examples.....	158
.....	158
.....	

.....	158
.....	158
<b>56:</b> .....	<b>159</b>
.....	159
.....	159
Examples .....	159
.....	159
git .....	159
.....	159
.....	160
.....	160
Windows .....	160
UnixLinux / OSX .....	160
.....	160
.....	161
.....	161
.....	161
.....	161
Windows .....	162
.gitconfig .....	162
.gitconfig-work.config .....	162
.gitconfig-opensource.config .....	162
Linux .....	162
<b>57:</b> .....	<b>163</b>
Examples .....	163
.....	163
<b>58: Git</b> .....	<b>164</b>
Examples .....	164
AtlassianSVNGit .....	164
SubGit .....	164
svn2gitSVNGit .....	164

Team FoundationTFVCGit.....	165
MercurialGit.....	165
<b>59:</b> .....	<b>167</b>
.....	167
.....	167
Examples.....	167
.....	167
.....	167
.....	167
<b>60:</b> .....	<b>168</b>
.....	168
.....	168
Examples.....	168
-MSG.....	168
.....	168
.....	168
.....	169
.....	169
.....	169
-MSG.....	169
.....	169
.....	169
.....	170
.....	170
Maven.....	171
.....	171
<b>61:</b> .....	<b>172</b>
.....	172
.....	172
Examples.....	172
Listmasterorigin / master.....	172
.....	173

---

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](mailto: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

```
git --version
```

Git。 [Git](#) 。

Git 。

### GitGit

```
git --version
```

.git **Git**◦

**Git**;

```
git --version
```

;**Git**

```
git --version
```

```
git --version
```

“”◦

.gitignoreadd.gitignore ◦

```
git --version
```

◦ ◦ ◦

-m◦

git remote add◦

git remote add

1. origin

2. URLhttps://<your-git-service-address>/user/repo.git

```
git --version
```

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

projectname**Git**◦ .git◦

MyFolder



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

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

1. ◦

2. ssh

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

httpsssh◦ **GitHub** httpsssh◦

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 -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
```

/

```
$ 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
```

◦

.git◦ ◦

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

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

ssh:◦

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

--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/.gitconfigWindows %USERPROFILE%\ .gitconfig °
```

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.

**repo**git config °

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

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.

°

.....

- 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.

## 2.8

- git

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.

```
user.nameuser.email
```

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.

**git - - --help**help°

```
git diff
```

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

```
status
```

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

```
-h
```

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

## GitSSH

**Windows**[Git Bash](#) ◦ **MacLinux**◦

**SSHSSH**◦

~/ .ssh

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

**SSH**◦

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

**BitbucketGitHub**[id\\_\\*.pub](#)◦

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

**EnterReturn**◦ ◦

**SSHssh-agent**◦ **ssh-agent**

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

**SSHssh-agent**◦ [id\\_rsa](#)

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

**HTTPSSH**

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

**SSH**

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

**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>

```
$ 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
```

GitGit

```
$ 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
```

Linux

LinuxGit。 Fedorayum

```
$ 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
```

DebianUbuntuapt-get

```
$ 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
```

Mac

MacGit。 GitSourceForge。

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

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

```
$ 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
```

+ svnGitSubversion8。

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

```
$ 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
```

## Windows

WindowsGit。 msysGit。 GitHubinstaller exe

```
$ 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
```

SSHGUI。

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

```
git shortlog -sn
```

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

[.mailmap](https://riptutorial.com/zh-TW/git/topic/1270/-mailmap-) <https://riptutorial.com/zh-TW/git/topic/1270/-mailmap->

---

## 3: Bash UbuntuGit

**bashgit** ◦ **git** ◦ ◦

### Examples

#### PS1

PS1Linux / 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 requireForcefalsegit clean-f-n-i -fGit.git
-i -	
-n - dr-run	
-q -	

## Examples

```
git clean -fX
```

o

```
git clean -fX
```

o

```
git clean -fd
```

o o

```
git clean -fd
```

o

```
git clean -f
```

o

```
git clean -i
```

```
git clean -i
```

iX do

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>`

<b>-p-u - patch</b>	
<b>-s - no-patch</b>	◦ <code>git show--patch</code>
<b>-</b>	<b>diff</b>
<b>--diff=</b>	◦ <code>myers minimal patience histogram</code>
<b>-</b>	
<b>--name</b>	
<b>--name</b>	<b>MAD</b>
<b>-</b>	◦ <code>core.whitespace</code> ◦ ◦ ◦ <b>--exit-code</b>
<b>--full</b>	<b>“”blob</b>
<b>--binary</b>	<code>--full-index diffgit apply</code>
<b>-a - text</b>	◦
<b>-</b>	<code>;;--color=alwaysdifflessgit</code>

## Examples

```
git diff
```

```
◦ ◦ git add git add ◦
```

```
git diff◦
```

```
git diff --staged
```

```
◦
```

```
git diff --staged
```

--staged

```
git diff --staged
```

status°

```
git diff HEAD
```

```
git diff HEAD
```

°

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

### 3

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

..°

°

## meld

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

°

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

### 2°

```
git diff myfile.txt
```

myfile.txt °

```
git diff myfile.txt
```

documentation/ °

HEAD

```
git diff myfile.txt
```

```
git diff myfile.txt
```

```
ada9b57my_branchnamemy_branchname my_changed_directory/
```

```
git diff myfile.txt
```

## word-diff

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

◦

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

word-diff

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

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

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

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

diff3

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

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

◦ secondfirstlast

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

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

```
git diff HEAD^ HEAD
```

◦

## Diff UTF-16plist

gitUTF-16os iOSmacOS◦

~/ .gitconfig◦

```
[diff "utf16"]
```

```
textconv = "iconv -f utf-16 -t utf-8"
```

iconv°

.gitattributes° ~/.gitattributes °

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

**git** .strings.strings°

°

**plist**.gitconfig

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

.gitattributes

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

**neworiginal**

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

**neworiginal**

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

**git diff**

**git diff original..HEAD**

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

° git --diff°

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

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

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

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

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

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

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

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

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

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。

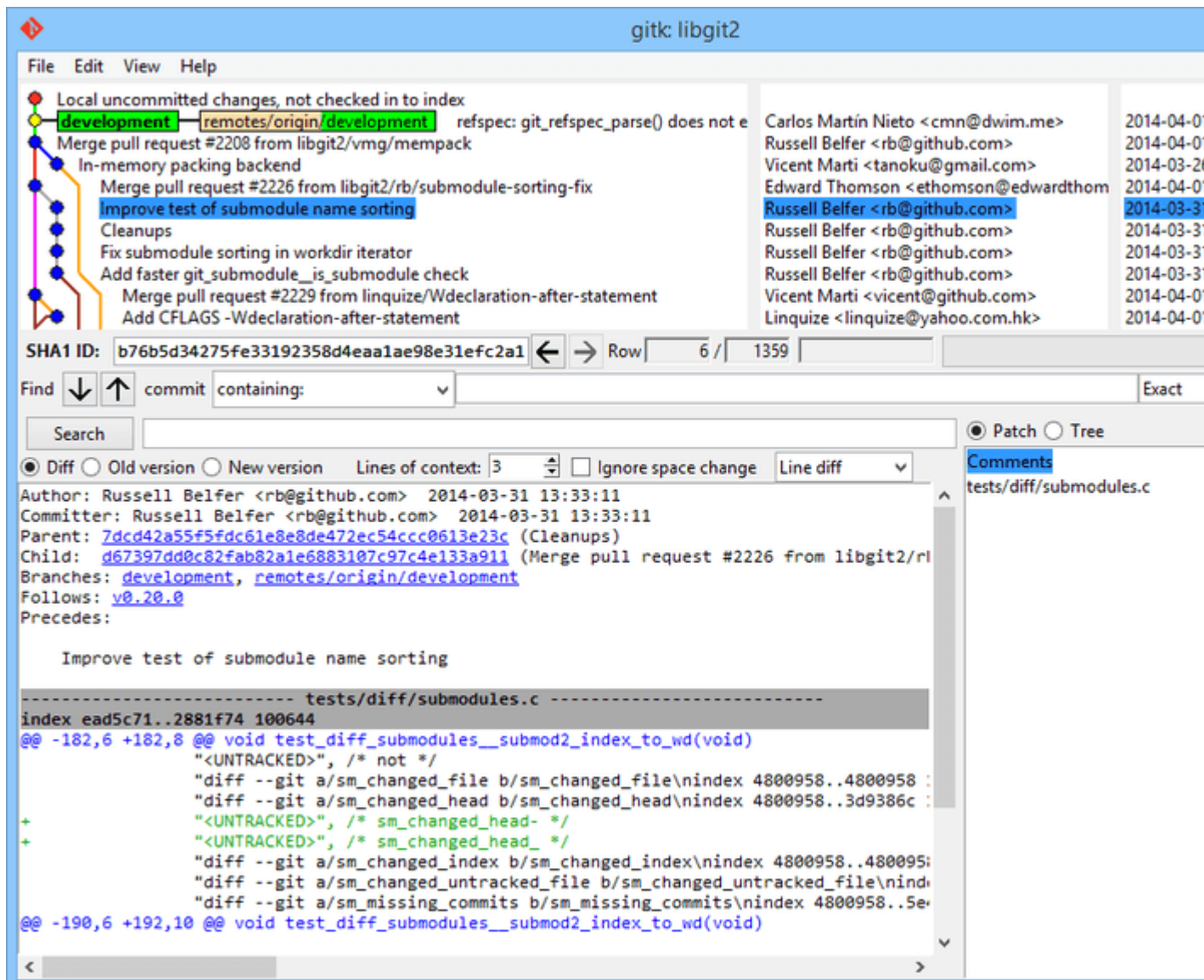
gitk。 GUI shellgit loggit grep。 。

Gitk。 cdGit

\$ gitk [git log options]

Gitkgit log。 --allgitkrefHEAD。 Gitk





1-1. gitk.

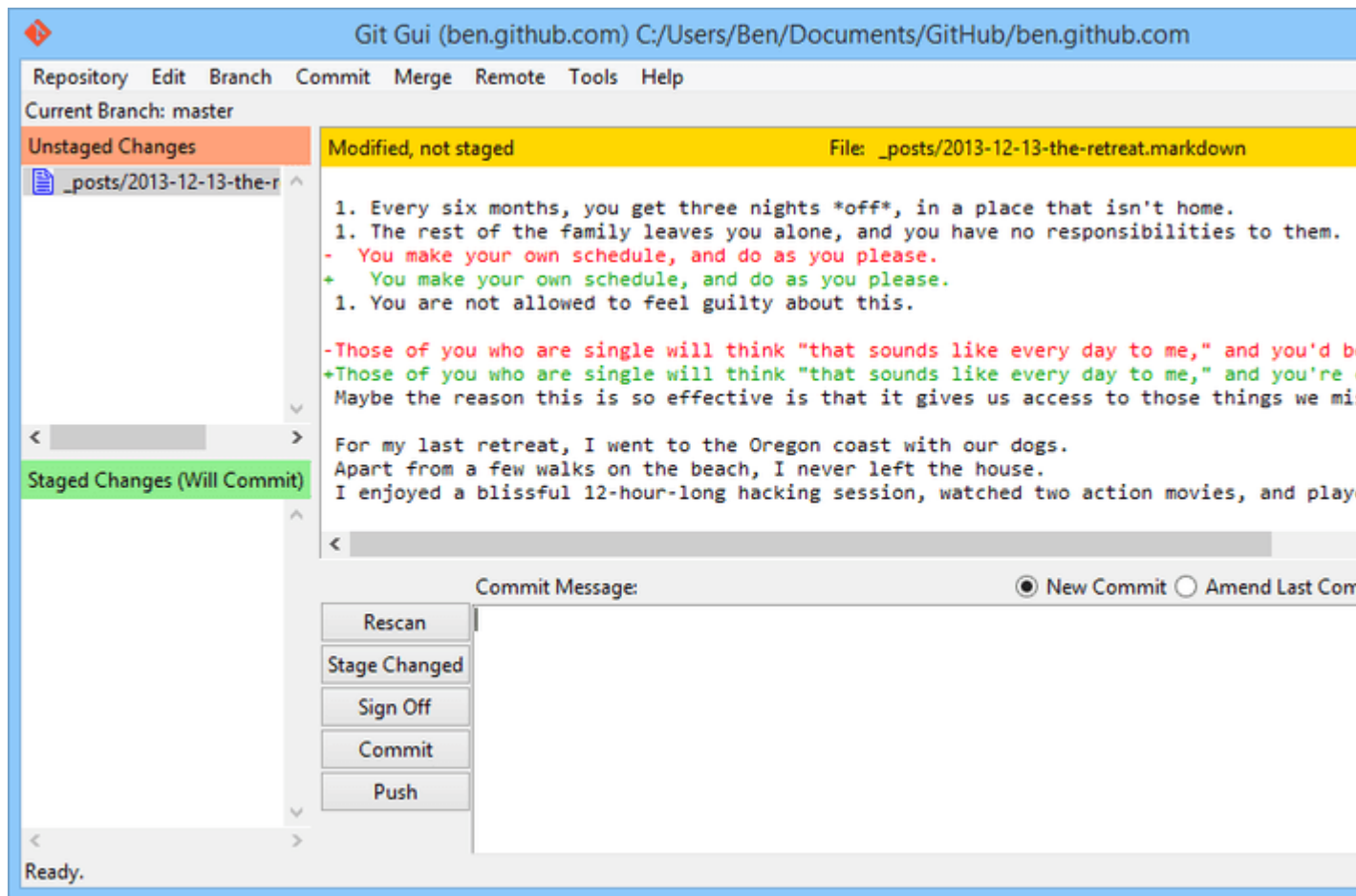
git log --graph;refs. HEAD. ;. .

git. .

git-gui.

\$ git gui

git-gui.



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](http://www.syntevo.com)

## Git

<https://gitextensions.github.io>

Windows



## 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.commitencodingUTF-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 -	◦ ◦
--resolve-msg = <MSG>	<msg>◦ --continue--skip◦ git rebasegit am◦
--abort	◦

# Examples

◦

1. ◦

2. `git format-patch <commit-reference> <commit-reference>`◦

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

2`HEAD~~`

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

`git apply some.patch.patch`◦ ◦

```
git am some.patch
```

```
git am some.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>`

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

-	

# 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
premium
premiumPro
origin
```

## GitURL

◦ URL

```
git remote set-url
```

2originupstreamurl。

```
git remote set-url
```

```
git remote set-url
```

```
git remote set-url
```

```
git remote show <remote repository alias>
```

```
git remote show origin
```

```
git remote show origin
```

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 - > - > - > - >

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

listserv

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

2

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

--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-svnMailingdevel@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 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/
```

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

\_\_\_\_\_

## SVN

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

## SHA1

```
git svn dcommitgit. git commitSVNSVN. git. git commitSHA1
```

git-svngitgitSHA1 IDgitSHA1gitSHA1。 SVNsvn。

SHA1/

## git svn rebase

```
git svn rebase
```

```
Checksum mismatch: <path_to_file> <some_kind_of_shal>
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

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

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 cloneSVNgit。 SVN。
```

```
gitmasterSVNtrunk。
```

## SVN

```
git pull
```

```
git svn rebase
```

SVN 。

```
git svn rebase
```

SVN。

## SVN

```
git svn dcommit
```

```
gitSVN。 SVNgitSVNgit svn rebase 。
```

gitgit repogit

- git add FILEgit checkout -- FILEgit checkout -- FILE /git checkout -- FILE
- git commit。 “”SVN repogit
- git stashgit stash popstashes
- git reset HEAD --hard
- git log
- git rebase -i
- git branchgit checkout

git-svn“SubversionGit”Subversiongit。

```
git//。  git rebasegit rebase 。
```

。 。“”SVN。

**git merge“”SVN** git merge commitsvn。

git。 git。 SVN。 git-svngit SVN 。

--rmdirSVN

```
git svn dcommit --rmdir
```

。

## dcommitgit GUISourceTreedefault

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

### .gitconfig

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

## SVNgit

```
git clean -fd
```

## SVNagaing SVN

```
git svn makedirs
```

## SVN

```
git clean -fd && git svn makedirs
```

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

# 13: GIT-TFS

Git-tfs GitTeam 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

kdifff3。

```
C:\> choco install kdifff3
```

Git。 Unix'NoAutoCrlf'。

```
C:\> choco install kdifff3
```

chocolateygit-tfs。

```
C:\> choco install kdifff3
```

### 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
```

```
/build  
app.js
```

Gitapp.js.

“build”“ .gitkeep”Git。

```
/build  
app.js
```

```
/build  
app.js
```

Gitbuild / .gitkeep。

Git。

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

# 15: Git

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

Git**gitrevisions7**。

- [ ] git describe v1.7.4.2-679-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 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'
```

## 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 /-ggit log`。

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

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

## / @{ }

```
$ 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>` 。

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

^

```
$ 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>^^^ 。

。

`git name-rev`

```
$ git reset --hard HEAD^        # discard last commit
```

```
$ git rebase --interactive HEAD~5    # rebase last 4 commits
```

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

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

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

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

**^ {/ }//**

```
$ 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 LFS Git LFS git

LFS; PSD JPEG; 3D

LFS GitHub <https://github.com/blog/1986-announcing-git-large-file-storage-lfs> ; Atlassian [git-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)
```

```
$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)
```

```
$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)
```

pre-push.sample git init ◦

```
$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)
```

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
<output follows>
v0.1
v1.3
```

## GIT

- `git tag < tagname >`

`tag °`

- `git tag < tagname >`

`tag commit-identifier°`

## GIT

- `git tag < tagname >`

- `git tag < tagname >`





# 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 shortloggit◦

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

/◦ 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

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

gnu locatemlocate **git dirs**

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

git shortlog

```
git shortlog -s
```

。

--all

```
git shortlog -s
```

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

---

## 20: Reflog - git

Git reflog HEAD ref. HEAD. reflog.

ref~30 reflog.

### Examples

#### rebase

rebase

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

rebase. git reflog

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

ddddddd HEAD@{n+1} *rebase*.

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

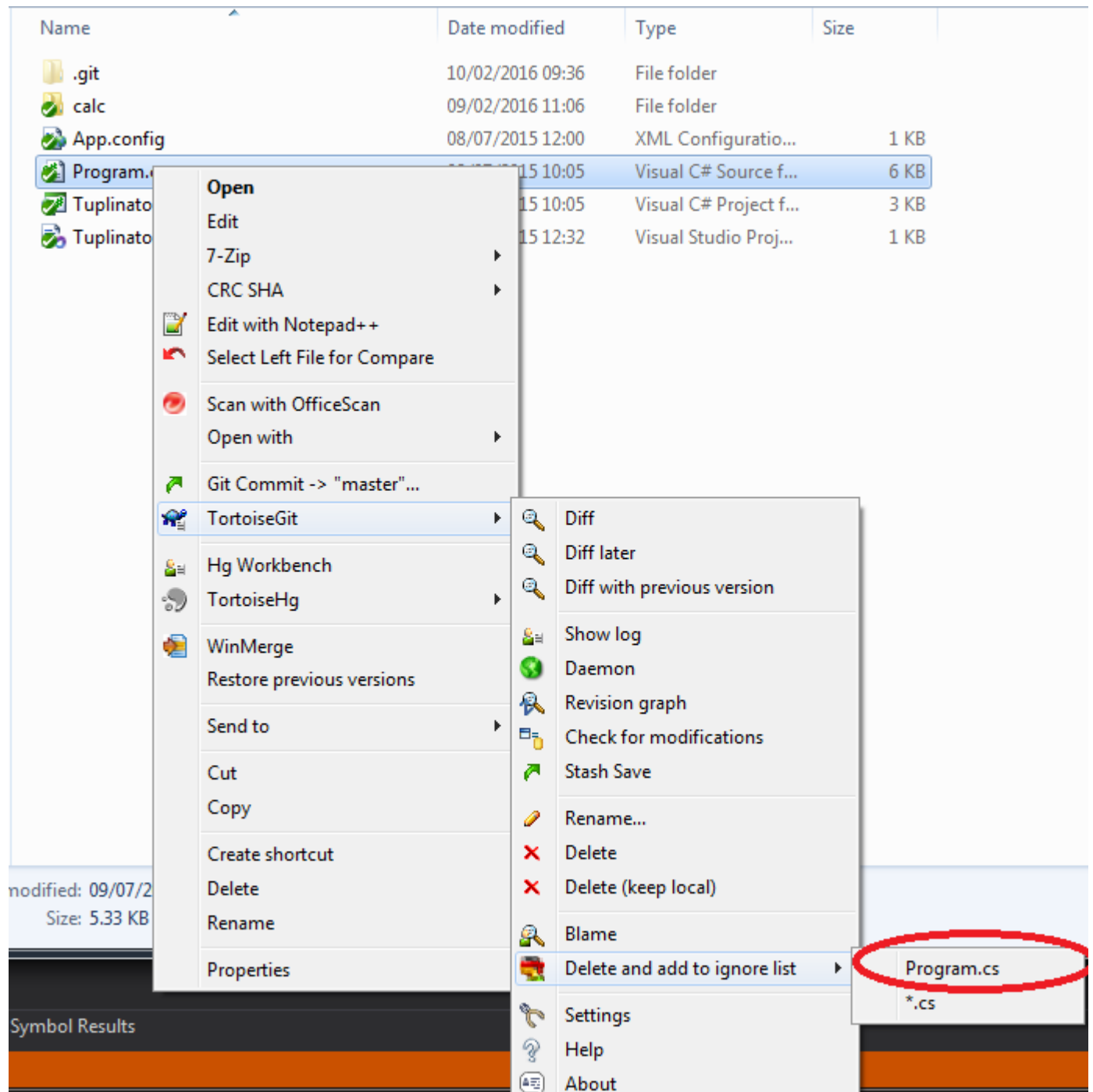
git checkout -b [branch]. .

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

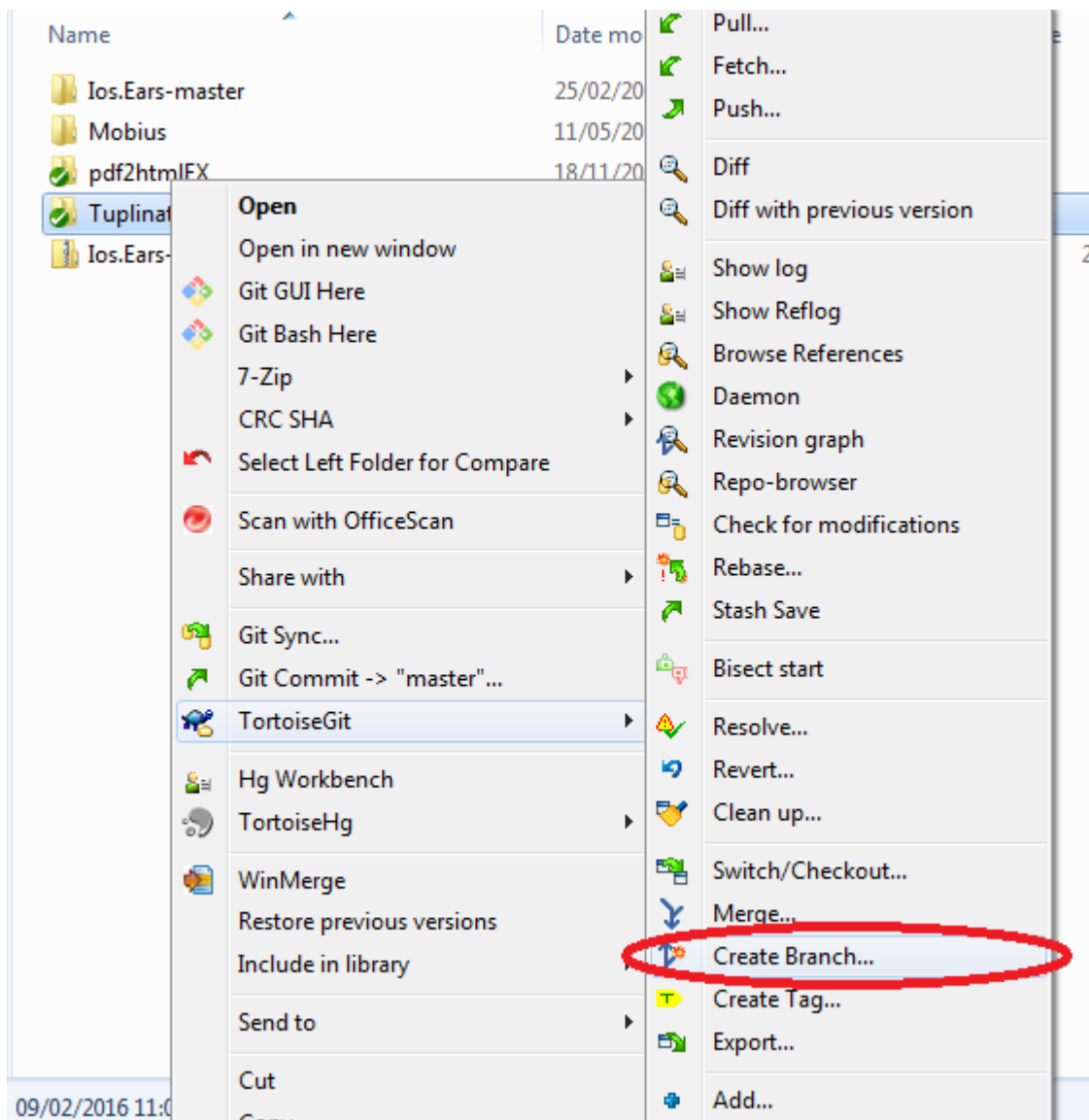
# 21: TortoiseGit

## Examples

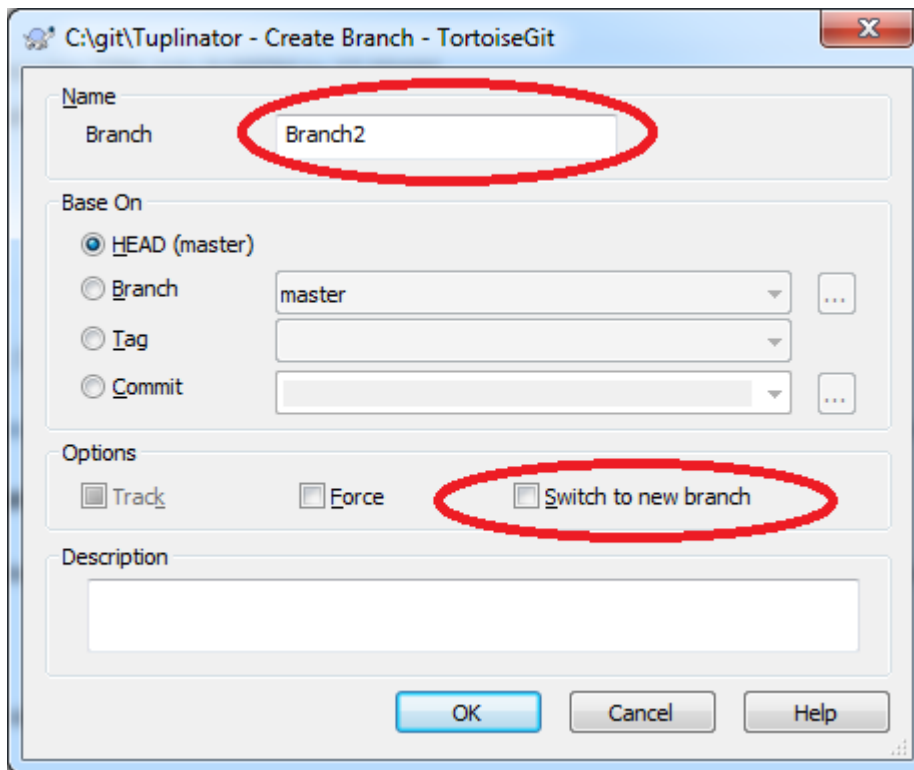
TortoiseGit -> TortoiseGit -> Delete and add to ignore list -> Ok.



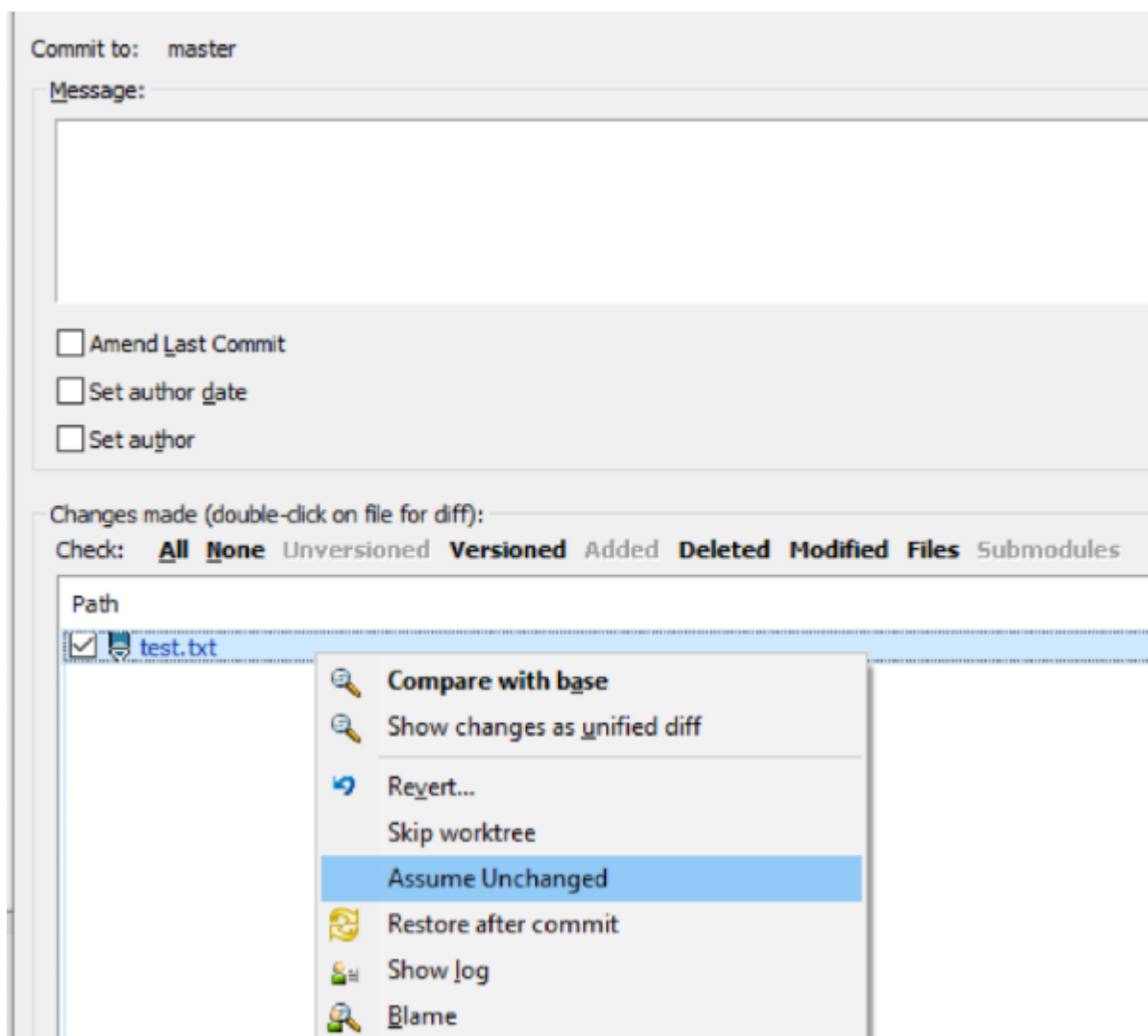
UI -> Tortoise Git -> Create Branch...



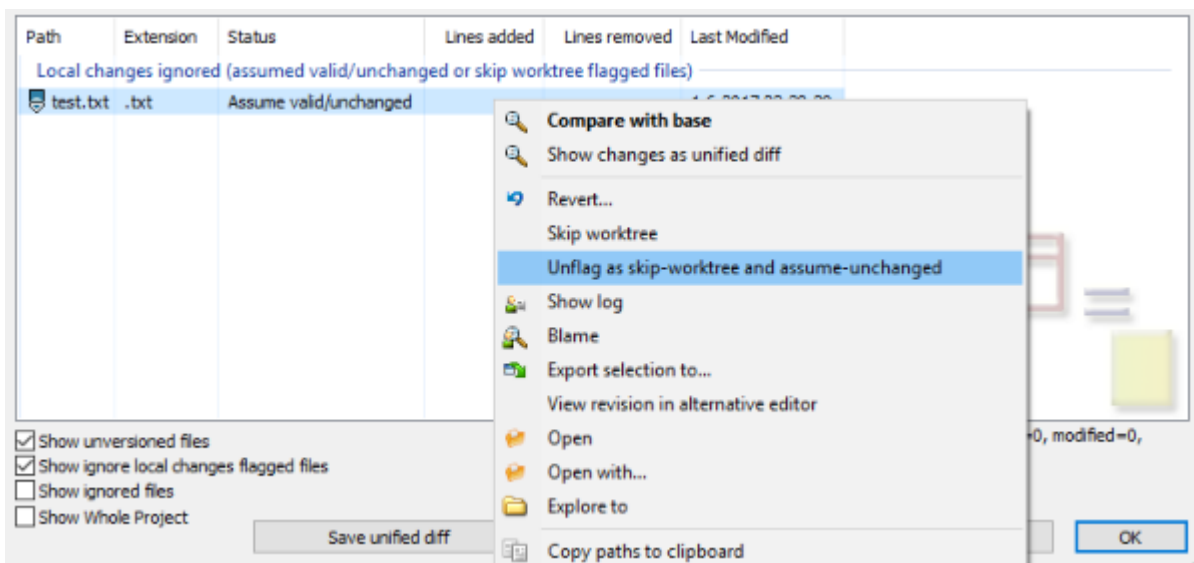
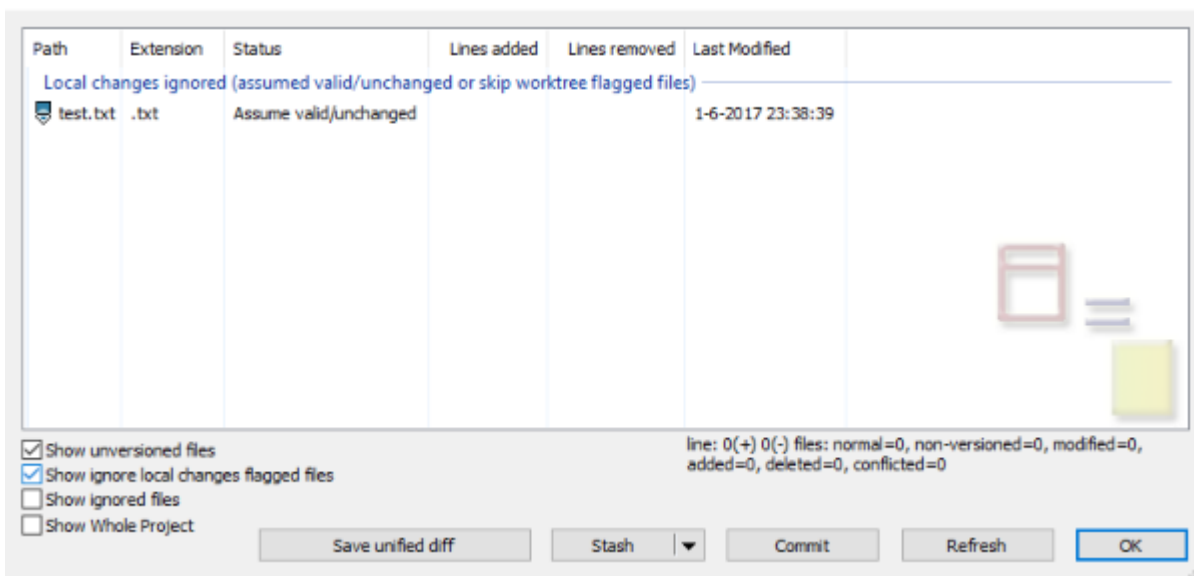
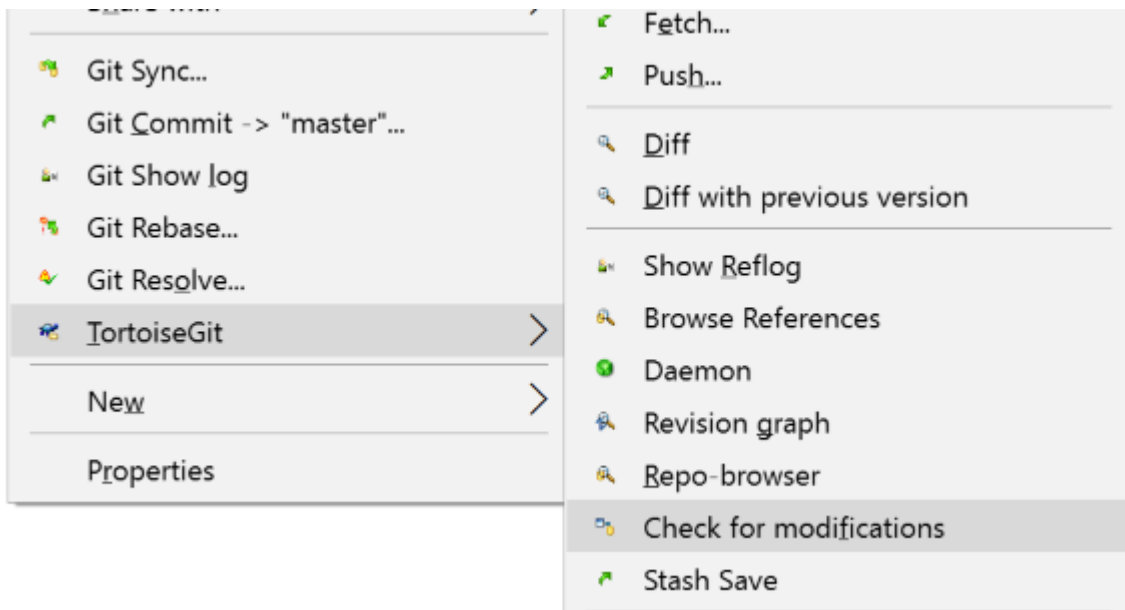
Give branch a name -> Give branch a name ->Switch to new branch。 ->OK。



“”



“”



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

Graph	SHA-1	Actions	Message
	00000000000000000000...		Working tree changes
	99fa32673e03741...	!	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 IncludeMdc
	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

Compare re

Revert chan

Combine to

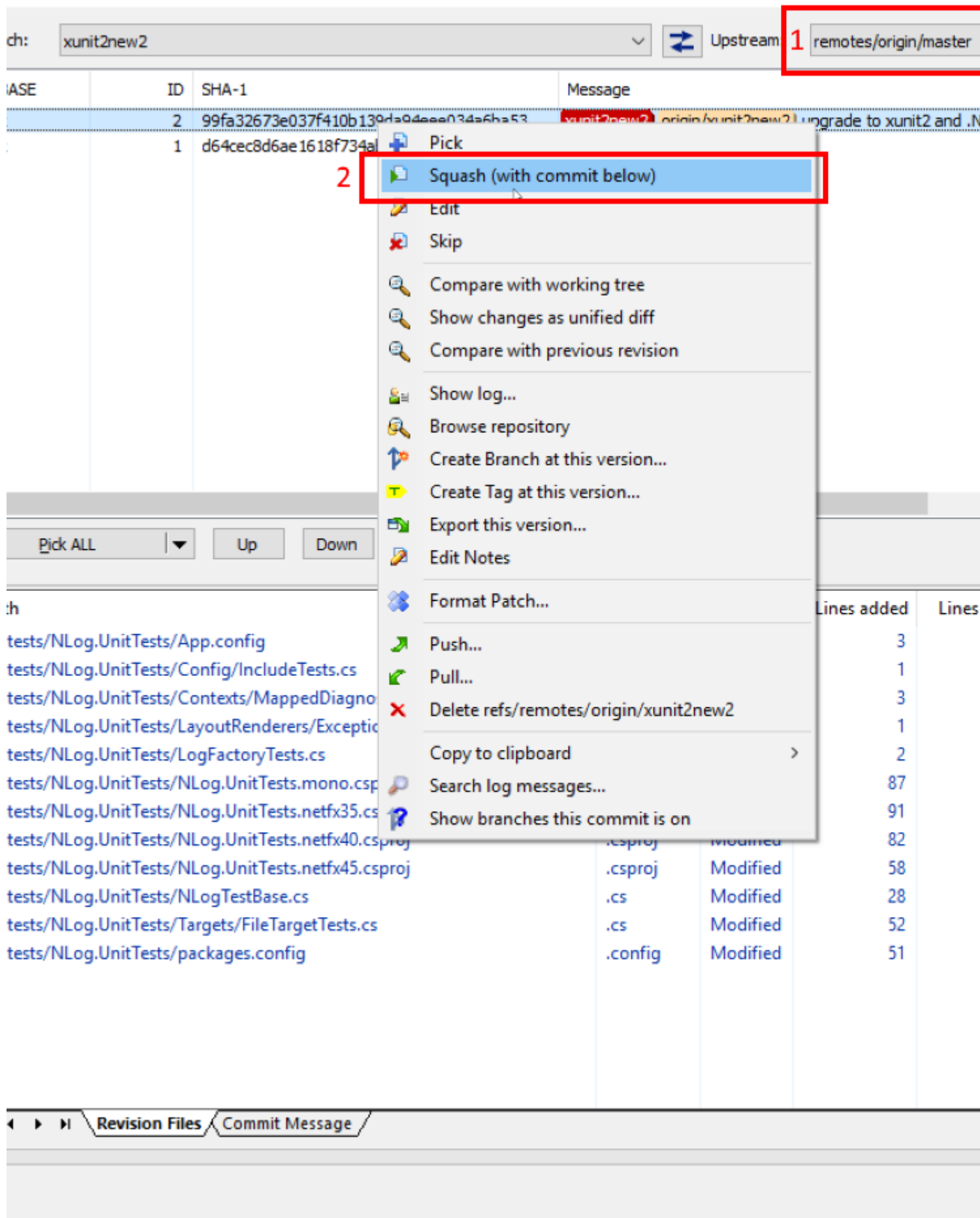
Format Pat

Copy to clip

Search log r

rebase





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]`

<b>-f --force</b>	<branch>◦ ◦
<b>-b</b> <new-branch> <b>-B</b> <new-branch>	<new-branch><branch> <new-branch>◦ <branch> HEAD ◦ -b◦ -B<new-branch><branch> ◦
<b>-</b>	<b>add</b> HEAD ◦
<b>- [ -</b>	<branch> --no-checkout◦
<b>-n --dry-run</b>	;◦
<b>-</b>	◦ <b>Git</b> ◦
<b>-v --verbose</b>	◦
<b>--expire</b> <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](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`

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

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

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

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

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

```
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<sub>input</sub>
- Windows<sub>true</sub>

Git<sub>◦</sub> .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

```
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
```

### 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` **git** `git-repository-url` `upstream`

“”

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

`pull` `fetch` `merge` ◦

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

`pull` **with** `--rebase` **flag** `fetch` `rebase` `merge` ◦

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

## LS-

`git ls-remote` / ◦

`refs / heads` `refs / 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 --ref
```

## Git

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

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

。

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

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

origin

```
git remote show origin
```

## URL

```
git remote show origin
```

## 2.7+config

```
git remote show origin
```

```
git remote
```

fetchpush URL

```
git remote
```

```
git remote
```

```
git push <remote_name> <branch_name>
```

```
git push origin master
```

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



## git checkout

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

## git push。

## URLset-url

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

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

## GitURL

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

## URL

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

## URL

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

```
git remote rename
```

```
git remote rename
```

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

```
git remote
# origin
```

## URL

```
git remote
# origin
```

```
git remote
# origin
```

```
git remote
# origin
```

=====

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>] [--disassociate] [--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 1
```

### 1.8.3

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

### 1.8.3

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

```
git clone <url>
```

◦

```
git clone <url>
```

◦

**URL** `--branch <branch name>`

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

`--branch-b` ◦ `<branch name>` ◦

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

`--single-branch` `[directory]` ◦ ◦

--single-branch

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

## 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  
ref: refs/heads/mainline
```

```
“” - HEADref object◦
```

```
ref◦ object◦
```

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

```
`git / refs / heads /◦
```

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

```
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>
committer 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 -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 d429810
package com.example.project

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

git commit

1. blobstrees - .git/objects
2. 1treecommit - .git/objects
3. .git/HEADHEAD **ref**commit

git ◦

## HEAD

hashrefgit checkout git ◦

1. committree
2. HEAD**ref**

git reset --hard**ref**shash / ref ◦

MyBranchb8dc53

```
$ 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 -A
```

2.x git add .。 1.x。

```
git add -Agit add --all git。
```

```
git rm filename
```

git--cached

```
git rm filename
```

```
git reset <filePath>
```

git add -i --interactive。 Gitgit 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.js44 " " +4/-4"

2. package.json" "

◦ 1-8 s u r a p d q h ◦

status◦

update◦

revertHEAD◦

add untracked◦

patchstatus◦

diff◦

quit◦

help◦

## hunk

patch"

```
git add -p
```

```
git add -p
```

◦

```
git add -p
```

- y
- n
- Q;
- []
- d
- g
- /
- j
- J
- k
- K
- []
- e
- []

◦

```
git add --interactive ◦
```

```
git diff --cached
```

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

---

30:

Git ◦ ◦

## Examples

### Gitflow

[Vincent Driessen](#)Gitflowgit ◦ ◦

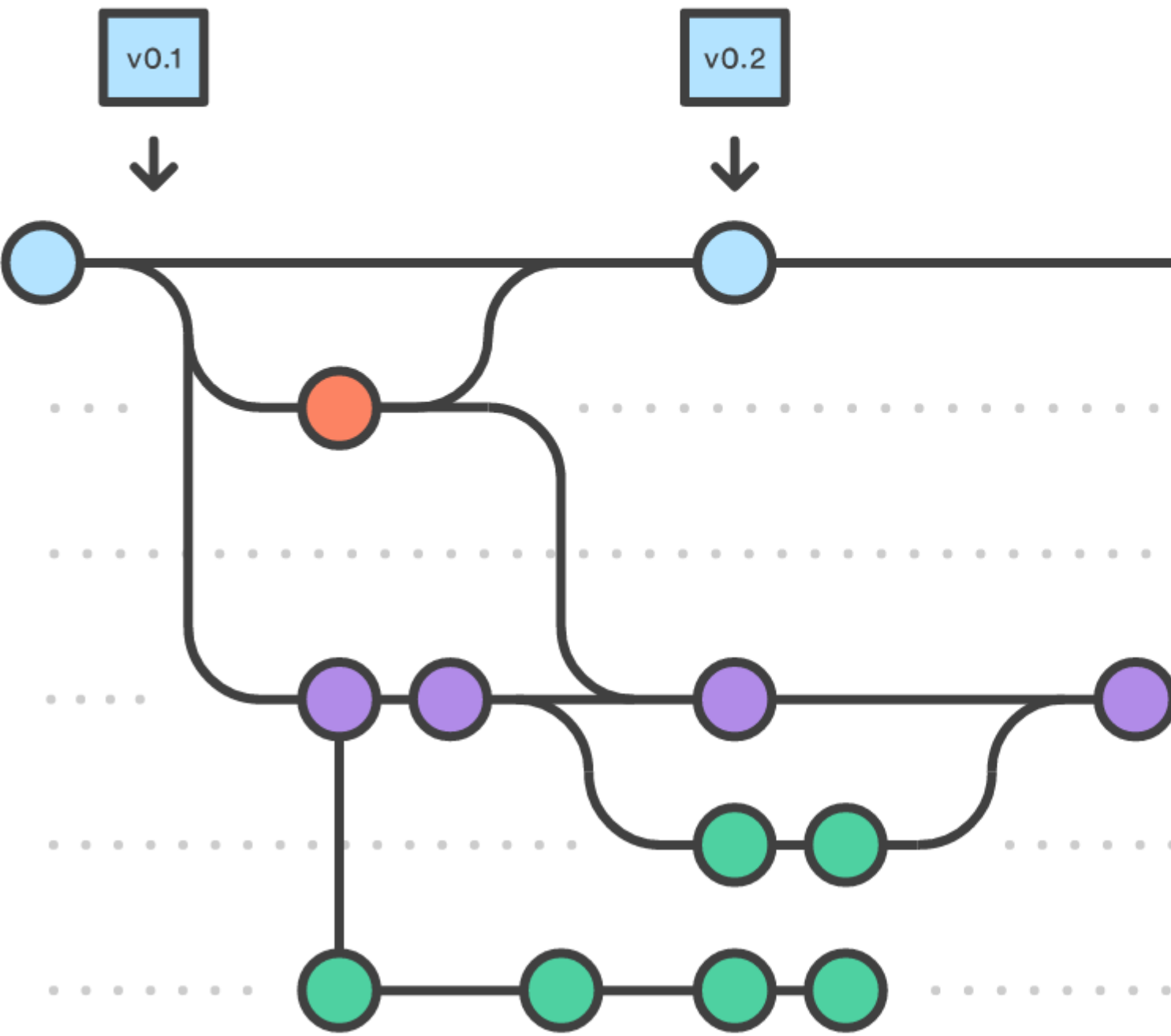
- master ◦ ◦
- develop ◦ ◦ /release ◦
- hotfix ◦ hotfixmaster masterdevelop ◦
- releasedevelopmaster ◦ masterdevelop ◦ master ◦
- feature ◦ develop ◦ feature ◦

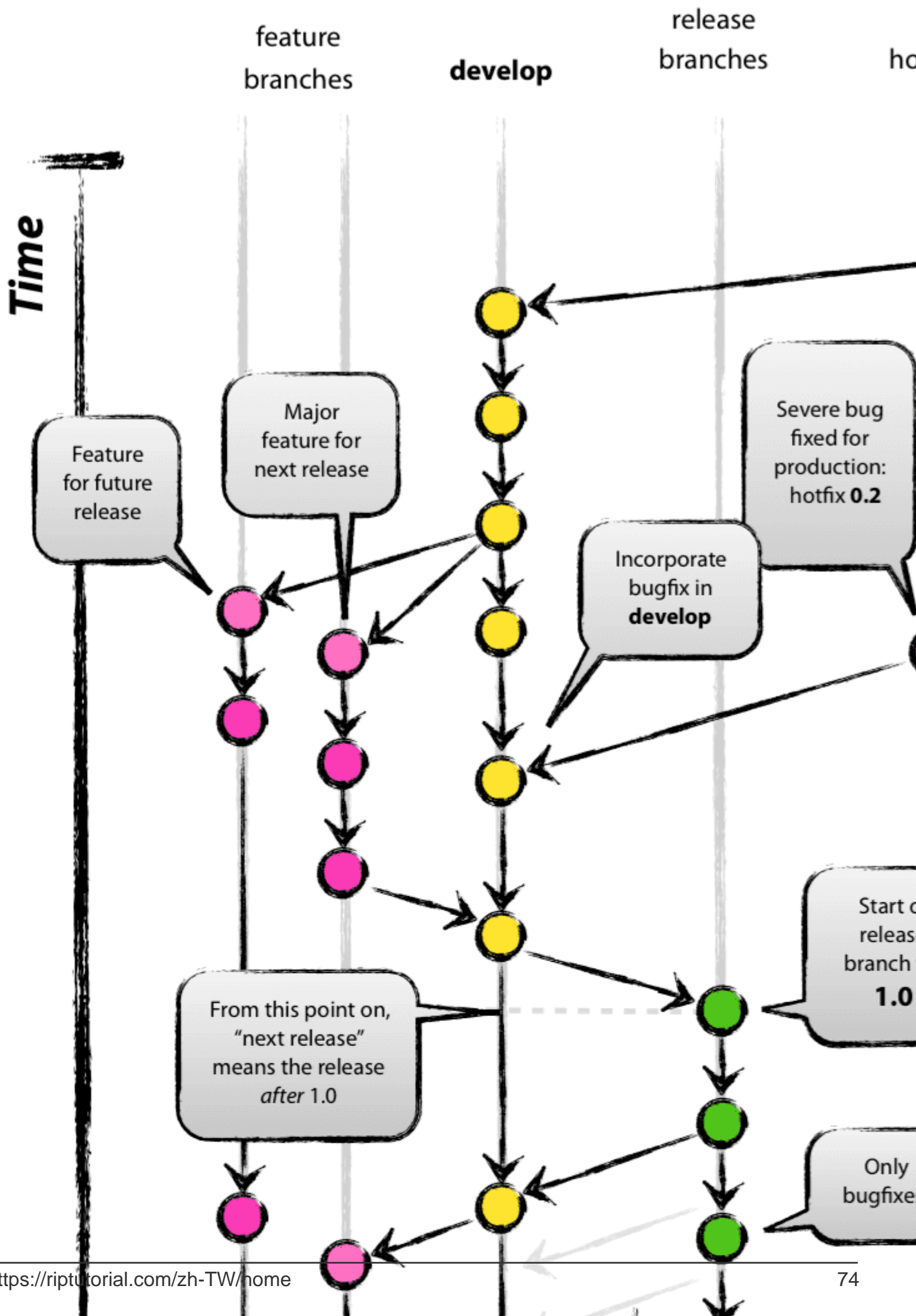
Master

Hotfix

Release

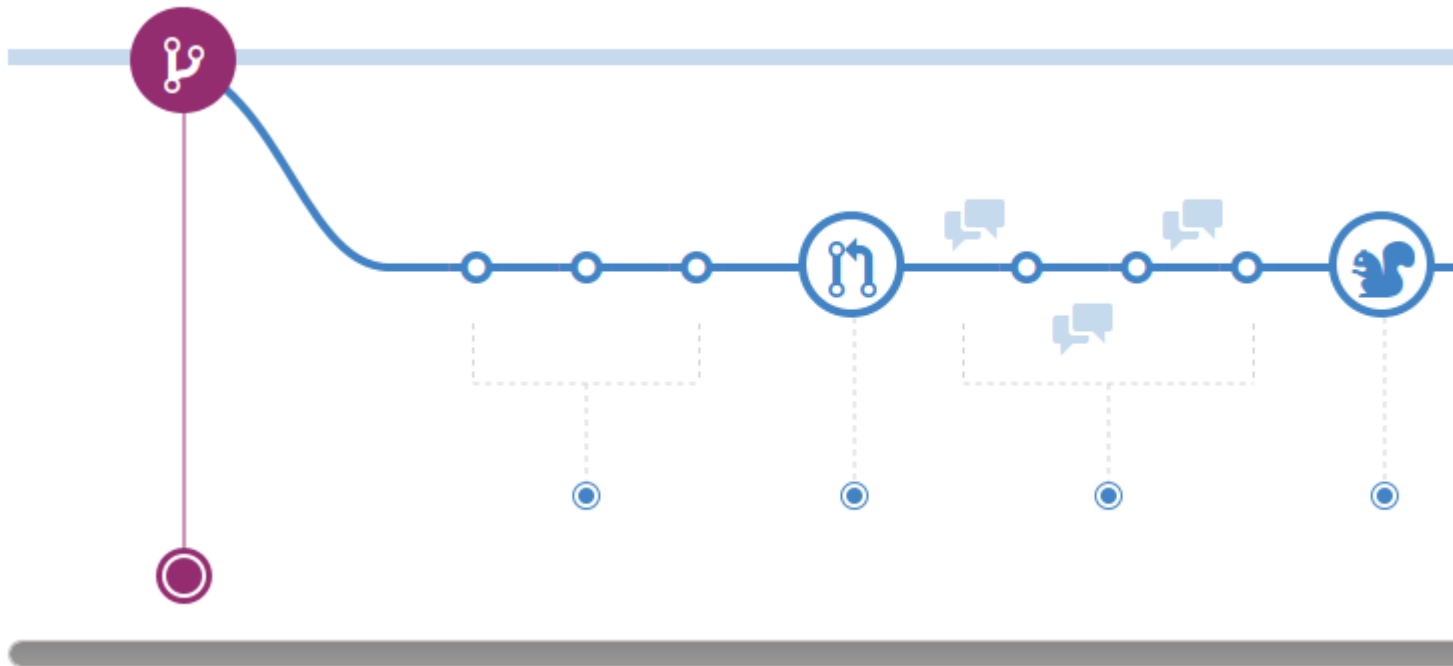
Develop





- 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 -a git 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 branch <name>
```

```
git branch <name>
```

## HEAD

```
git branch <name>
```

<start-point>**git SHAHEAD**

```
git branch <name>
```

<remote\_name>**origin**

```
git branch <name>
```

```
git branch <name>
```

```
git branch <name>
```

◦ **“”**

```
git branch <name>
```

```
git branch <name>
```

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

dev◦ `dev$ git branch -d`

```
$ 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 new_branch_name
```

°

file.example path/to/ °

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

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

--° --°

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

some-file°

origin**Git 1.5.0**

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

**Git 1.7.0**

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

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

◦

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

```
git push origin :<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 <REMOTENAME> <BRANCHNAME>
```

```
git◦ new-feature new-feature ◦ :
```

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

## HEAD

◦ aabbcc

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

◦ ◦ [reflog](#)◦ ◦

◦

```
git checkout -
```

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

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

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

# 32:

## Examples

### Git

- ~/.gitconfig

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

- 

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

-

- git cigit commit
- git stgit status
- git cogit checkout ◦

### git◦

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

/

--get-regexp **git**

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

[alias].gitconfig

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

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

Gitgit<sup>sh</sup> shell! ◦

~/.gitconfig

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

## shellshell

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

f° git ignore .tmp/.tmp/.gitignore°

Git\$1 \$2° Git°

!git checkout° cd°

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

-

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

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

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

-

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

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

```
[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
```

◦

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

## .gitignore

```
[ alias ]

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

## grep

```
[ alias ]

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

```
[ alias ]
```

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

## Unstage

```
git reset commit reset◦ gitresetreset◦
```

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

```
git unstagegit 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 incomingBranchmaster ◦
```

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

```
git merge incomingBranch
```

## 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**◦ master◦ master◦ **PR**master◦

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

2. refoldvaluenewvalueref◦

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

oldvaluemasternewvalue ◦

<oldvalue>-d<ref> <oldvalue> ◦

--create-reflog update-refrefreflogreflog◦

-zNULupdatecreatedeleteverify◦

<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`

-	o
--abort	rebaseHEAD. rebaseHEAD. HEADrebase.
--keep	o
-	o
-m - merge	rebase. rebase. rebase. o o
--stat	rebase. diffstatrebase.stat.
-x - exec command	rebasecommand

rebase.

o o

## Examples

o

rebaserebase.

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

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

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

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

rebase. o git pushgit pushgit push --force.

## Rebase

rebase“”“”

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

HEAD“”

rebaseHEADmaster ;cherry-pickingtopictopic°

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

/local master remote topic

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

---

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

topic

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

rebaserebase

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

git rebase upstreamgit rebase upstreamHEAD“”“”“”°

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

rebase“” topic“”

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

## Rebase

git rebase ° git rebase °

## rebase

```
git rebase -i
```

-i ◦ **rebase/**◦

◦

```
git rebase -i
```

◦ ◦ **rebase**◦ git log◦

---

◦ ◦

```
git rebase -i
```

pick reword◦

**rebase**◦ ◦ ◦

---

◦ pickedit◦ **Git**◦ ◦

◦ ◦

---

◦ pickedit**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
```

```
$ 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
```

```
$ 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
```

git.

```
$ 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
```

.

```
$ 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
```

rebase

```
$ 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
```

```
$ 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
```

```
$ 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
```

git rebase /.

## rebase

```
$ 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
```

◦ ◦ git rebase◦

## git-pullrebase

rebasegitgit pullrebase◦

.gitconfig .gitconfig.git/config

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

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

git pull--rebase

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

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

## rebase

pull◦ -x◦

git rebase -i -x make

rebasemake◦ makegit◦

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

## KDiff3

.gitconfig

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

pathKDiff3

## 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
```

cmdo IDEProgram 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")\"
```

cmdo IDEProgram 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 updategit submodule update ◦

### Git

◦

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

◦ git submodule update --init --recursive ◦

◦

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

◦

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

```
git pull
```

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

◦ git status◦

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

```
git pull git pull --rebase◦ ◦
```

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

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

### SHA1“gitlink”

◦

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

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

## SHA1

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

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

## 1.8

the\_submodulethe\_submodule

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

- ```
git submodule deinit the_submodule.git / configthe_submodule s' git submodule update git submodule syncgit submodule foreachthe_submodule ° ° git submodule initgit 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. 

```
.gitmodulesgit 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`

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

8. 。

---

[Axel Beckett](#) [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 remote add plugin https://path.to/remotes/plugin.git
```

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

## Backport

1.

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

2.

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

3. Cherry-pick backports

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

4.

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

🔗 <https://riptutorial.com/zh-TW/git/topic/1634>

- `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`。

```
# 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`。 `git`。

`git bisect reset` `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
```



```
# 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
```

[[script]] [[script]] [[arguments]] [[...]]。

[[...]]git bisect good [[...]]git bisect bad [[...]]。 [[...]]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 start master old-rel
```

[[...]]

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

。 [[...]]git reset [[...]]git [[...]]。

[[...]]git [[...]]。

[[...]]bisect [[...]]

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

[[...]]git [[...]]。

[[...]]/[[...]] [https://riptutorial.com/zh-TW/git/topic/3645/\[\[...\]\]-\[\[...\]\]](https://riptutorial.com/zh-TW/git/topic/3645/[[...]]-[[...]])

40:

Git。 .git/exclude git update-index --assume-unchangedgit update-index --skip-tree Git 。

## Examples

.gitignore

.gitignoreGit - Git。

.gitignore/。 .gitignore

1. 。
- 2.
- 3.

。

1. Git
2. git statusgit diff
3. git add -A

。 Git。

.gitignoreglob

```
# 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/
```

```
# 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



```

/* .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)
\#*#

```

000 -X 0000 0000000000。 00-x 000000000000000000。

```
git clean
```

## Git

```
.gitignore
```

[illegible]

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

```

00000000Git00000000.txt0000000important.txt0000.

```

□ □

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

000000000000.txt0000000000。

```

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

```

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



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

git .gitignore 忽略 \*.o 文件，git check-ignore 忽略 Readme.md。

git .gitignore 忽略 git check-ignore -v

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

Git 1.7.6 使用 git status --ignored 查看被忽略的文件。使用 git .gitignore 查看被忽略的文件。

gitignore 文件

gitignore 文件

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

examples 目录下的 output.log 文件被忽略，src 目录下的文件被忽略。

gitignore 文件。使用 gitignore 文件。

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

src 目录下的 .gitignore 文件被忽略。

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

gitignore 文件

foo.txt 文件被忽略

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

gitignore 文件。使用 gitignore 文件。

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

在bar/目录下创建.gitignore文件。在bar/目录下创建.gitignore文件

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

在bar/目录下创建.gitignore文件

.gitignore文件在bar/目录下创建.gitignore文件。在bar/目录下创建.gitignore文件。

在bar/目录下创建.gitignore文件。在bar/目录下创建.gitignore文件。

在bar/目录下创建.gitignore文件

```
# 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/
```

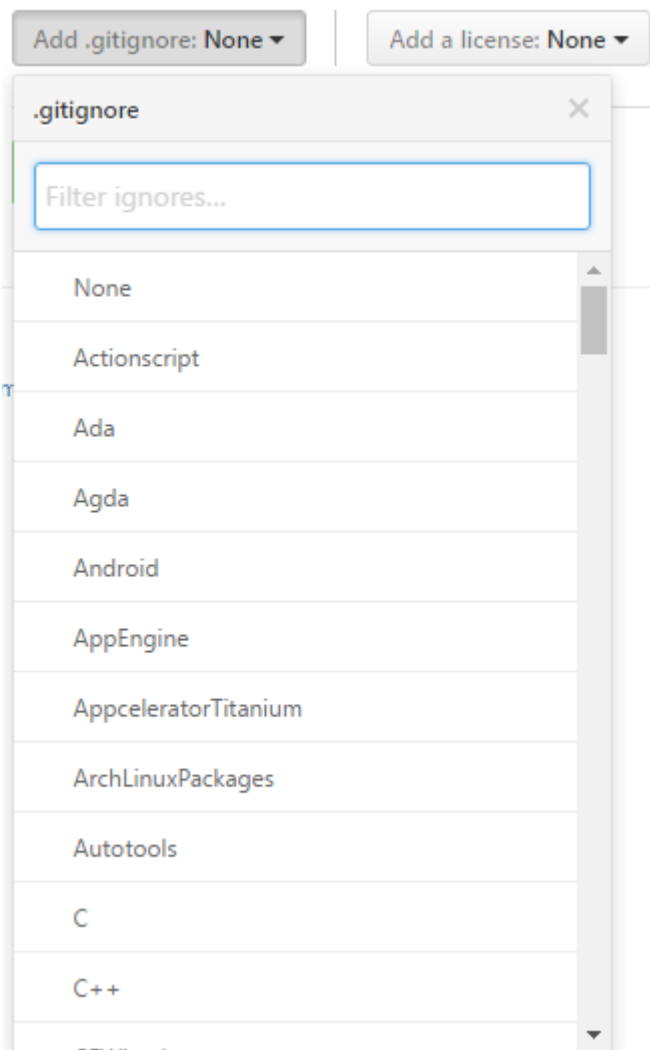
在bar/目录下创建.gitignore文件

在bar/目录下创建.gitignore文件。在bar/目录下创建.gitignore文件。

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

在bar/目录下创建.gitignore文件。在bar/目录下创建.gitignore文件。





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

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

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

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

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

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

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

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

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

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

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

```
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.config "nocommit"

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

.git/info/attributes.gitmodules

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

NOCOMMITGit。

- Windows。
- Git。。
- Windows

。 [ ]

.gitignore.git/info/exclude。

ignoreupdate-index

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

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

git hide git unhidegit hidden

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

--assume-unchangedupdate-index

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

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

git --assume-unchanged 告诉 Git 别跟踪这个文件。以后如果这个文件被修改了，Git 也不会提醒。但是，如果你以后又想跟踪这个文件，你可以用 git update-index --track 来告诉 Git。

git --skip-worktree 告诉 Git 别跟踪这个文件，但如果你以后又想跟踪这个文件，你可以用 git update-index --track 来告诉 Git。Skip-worktree 和 --assume-unchanged 的区别是，Skip-worktree 会阻止你提交这个文件，而 --assume-unchanged 不会。

创建 .gitignore 文件

git 使用 .gitignore 文件来跟踪哪些文件应该被忽略。你可以在你的项目根目录下创建一个 .gitignore 文件，里面列出你想要忽略的文件名。git 会自动忽略这些文件。

使用 git rm --cached 来删除已经提交的文件

```
# 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
```

创建 .gitkeep 文件

Git 使用 .gitkeep 文件来跟踪空目录。你可以在你的项目根目录下创建一个 .gitkeep 文件。

创建 .gitkeep 文件

gitkeep 文件是一个空文件，用于告诉 Git 这个目录应该被跟踪。你可以在你的项目根目录下创建一个 .gitkeep 文件。在 Windows 上，你可以使用 git bash 来创建这个文件。

```
$ touch .gitkeep
```

创建 .gitkeep 文件

创建 dummy.txt 文件

hack 使用 .gitkeep 文件来跟踪空目录。你可以在你的项目根目录下创建一个 .gitkeep 文件。在 Windows 上，你可以使用 git bash 来创建这个文件。

创建 .gitignore 文件

使用 git status --ignored 来查看被忽略的文件

```
git status --ignored
```

创建 .gitignore 文件

```
git status --ignored
```

创建 .gitignore 文件

```
git status --ignored
```

创建 .gitignore 文件

```
git status --ignored
```

```
 - - --untracked-files=all
```

```
git status --ignored
```

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

00 41: 00

## Examples

10/10

[illegible]

0000000000

```
git reflog
```

10/10/2019

[illegible]

XXXXXXXXXXXX ID.

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

[illegible]

□ □ □ □ □ □ □ □ □ □

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

81eeccf ID

10/10/10

```
00000000000000000000reset。
```

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

```
0000000000000000000000000000000000--hard00
```

Page 10 of 10

□ □

```
git reflog
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```
git reflog
```

```
git - 。
```

Page 10 of 10

## Git



git stash apply

```
git stash apply
```

git stash apply

```
git stash apply
```

git stash apply

<https://riptutorial.com/zh-TW/git/topic/725/>

42:

Git 的 pull 命令是 fetch 和 merge 的快捷方式。它默认使用 fetchmerge / rebase 策略。

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

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

git pull 等同于 git fetch 后跟 git merge。

## Examples

git pull

svn update 与 subversion 类似

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

2.9

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

2.9

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



```
git stash pop
```

1111111111

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

□ □ □ □ □ □

```
git pull
```

Page 10 of 10

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

```
reset --hard reflog reset reflog。
```

```

origin master

```

□ □ □ □ □ □ □ □ □

Page 10 of 10

```
00000000000000000000000000000000git0000000000000000。00000000000000000000Git 0000000000000000。
```

```
git pull --rebase
```

□ □

```
git pull --rebase
```

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

```
git pull --rebase
```

1

```
git pull --rebase
```

□ □ □ □ □ □ □ □ □ ○

11/11/2019

```
git pull --rebase
```

[illegible]

□□ "□□□□□"

```
.git..... .git..... .git.....
```

Page 10

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

10/10

1111

~~~~~ GitHub ~~~~~。 ~~~~~。

```
git pull
```

11/11/2019

[illegible]

```
git pull
```

```
[[{"feature-A": "origin", "URL": "https://www.google.com", "SHA": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0u1v2w3x4y5z6"}]]
```



```
git pull git fetch git merge
```

```
git pull
```

[illegible]

```
git pull
```

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □

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

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

git ssh rsync http

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

Cherry-pick

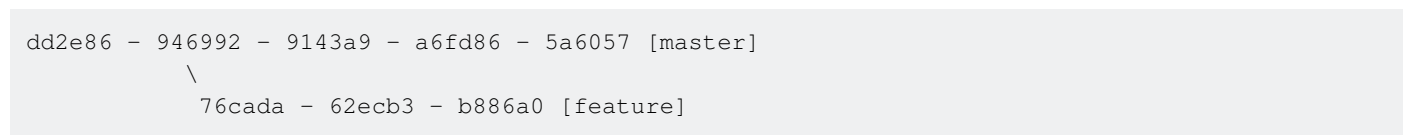
Git SCM Book

- 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

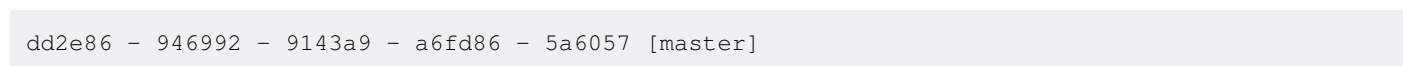
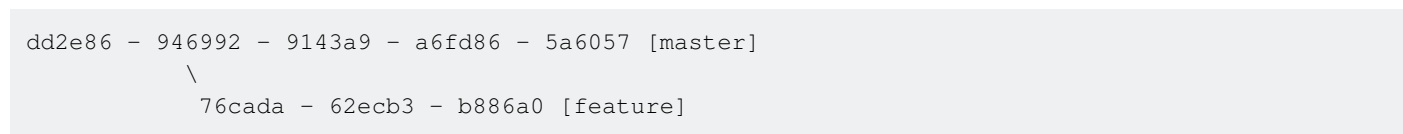
| -e -    | git cherry-pick     |
|---------|---------------------|
| -X      | "".                 |
| --ff    | HEADcherry-pick'ed. |
| -       | .git / sequencer.   |
| -       | .                   |
| --abort | .                   |

## Examples

git cherry-pick <commit-hash>



b886a0master5a6057



```
\
76cada - 62ecb3 - b886a0 [feature]
```

aaaaa66b23aaaaaaaaDIFFaaaaaaab886a0 aaaaaaaa。 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaab886a0 aaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaa。

aaaaaaaaaaaaaaaaaaaa

git cherry-pick <commit-A>..**<commit-B>**aaaaaaA aa aaaaaaaBaaaaaaaaaaaa。

git cherry-pick <commit-A>^..**<commit-B>**aaaAaaaaaaaaaaaaaaaaaaaaaaB。

aaaaaaaaaaaa

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa。

git branch --contains <commit>aaaaaaaaaaaaaa。

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

aaaaaaaaaaaaaa

aa git cherryaaaaaaaaaaaaaa。

aa

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

...aaaaaaaaaaaaaa

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

+aaaaaaaaaaaaaa developmentaaa。

aaa

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

aaa

**-v**aaaSHA1aaaaaaaaaa。

**<upstream>**aaaaaaaaaaaaaaaaaa。 aaaHEADaaaaaa。

**<head>**aaaaa;aaaHEAD。

**<limit>**aaaaaaaaaaaaaaaaaa。

aa git-cherryaaa。

aaaaaaaaaaaa <https://riptutorial.com/zh-TW/git/topic/672/aaaa>

45:

Git。Git。

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

|               |     |
|---------------|-----|
| -             | o o |
| --verbose     | o   |
| <>            | o   |
| <Refspec> ... | o   |

“ ”。 “ ”。

“ ”。 。

“ ”。 “ ”。 “ ”。

## Examples

```
git push
```

Git 2.xGit 1.x。

git。

```
git push
```

feature\_x



```

git push
git /

```

```
git push
```

```
git 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`

0000000000GitHub0

\_\_\_\_\_

[illegible]

$\frac{1}{n} \sum_{i=1}^n x_i = \bar{x}$

[illegible]

```
git push -f
```

\_\_\_\_\_

[illegible]

- [illegible]

\_\_\_\_\_

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

11

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

```
00000000 wip-yourname000000000000000000.
```

[illegible]

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

11

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

```
00000000 wip-yourname
```

```
--delete
```

11

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

[illegible]

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

1

```
git
```

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

000 347d700000000000000000

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

Page 10 of 10

**Current**

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

[illegible]

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



git config push.default current。

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

git config push.default current

git config push.default current

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

git config push.default current。

git

```
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 commitGitvimemacs ◦ -m

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

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

-m

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

Git。

git

git commit --amend

```
git commit --amend
```

git commit --amend

git commit --amend

git commit --amend

```
git commit --amend
```

git commit --amend

```
git commit --amend
```

git commit --amend

```
git commit --amend
```

git commit --amend

```
git commit --amend
```

git commit --amend

git commit --amend

git commit --amend

git commit --amend

```
git commit -a
```

git commit -a

```
git commit -a
```

git commit -a

```
git commit -a
```

git commit -a

```
git commit -a
```

git commit -a

```
git commit -a
```

git commit -a



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

git add --all git add --all



Repo-Cleaner” <https://rtyley.github.io/bfg-repo-cleaner/>.

bfgr --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 Smith <johnsmith@example.com>"
```

“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 commit file1.c file2.h
```

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

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

1000000000

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

[illegible][illegible]

git

1. 00000000000000
2. 00000000 500000
3. 000000
4. 000000000000
5. 00000000 000000
6. 00000000000000 720000
7. 000000000000000000

Chris Beam 7000 .

□ □ □ □ □ □ □

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

```
--date[YYYYMMDD] . [YYYYMMDD]git log[YYYYMMDD].
```

□ □ □ □ □ □ □ □

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

```
date[] [] GNU[] [] [] [] [] [] [] [] [] []
```

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

[illegible]

□ □ □ □ □ □ □ □ □ □ □ □ □ □

[illegible]

```
git add -p
```

11

```
git add -p
```

[illegible]

```
git add -p
```

□□□□□□□□□□□□□□。 □□□□□□□□□□□□□□□□

```
git add -p
```

。 。

```
git add -p
```

println /  
logging。

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

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

```
git config user.name "Full Name"  
git config user.email "email@example.com"  
  
git commit --amend --reset-author
```

GPG

## 1. 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]
```

ID 16。

## 2. git 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]
```

## 3. 1.7.9 git commit -S GPG

```
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]
```

🔗 <https://riptutorial.com/zh-TW/git/topic/323>



00 47: 00

## Examples

1111

11/11/2019

[illegible]

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

[illegible]

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

[illegible]

□ □ □ □ □ □ □ □ □

```
add-gremlins
```

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

[illegible]

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

```

00000000000000000000000000000000. 000000000. 00add-gremlins000000000000000000000000.

```

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

[illegible]

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

```
reflog
```

```

#####rebase#####commit#####pre rebase#####. #####reflog#####90##### - #####
#####

```

```
$ 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
...
```

```

##### rebase HEAD@{3}#####

```

```
$ 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
...
```

```

##### /##### /### rebase:

```

```

[[ref]log[[100[[

```

```
git reset --hard HEAD@{3}
```

```
#####git#####.
```

[illegible]

```
git checkout 789abcd
```

```
git checkout 789abcd
```

[illegible]

```
git checkout 789abcd
```

```
git checkout 789abcd
```

```
git checkout 789abcd
```

```
#####reflog#####
#####. git stash: git reset#####git reset --hard#####
```

```
git checkout -- file.txt
```

```
git checkout -- file.txt
```

```
git checkout -- file.txt
```

```
git checkout -- file.txt
```

```
git checkout -- file.txt
```

```
git stash -p git stash。stash pop stash  
drop stash drop。
```

```
git revert HEAD~1
git push
```

```
git revert HEAD~1
git push
```

```
git revert HEAD~1
git push
```

```
git rebase -i <earlier SHA>
```

135





```
> 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

```
> 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
```

**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 log --oneline --decorate
ccc3333 (HEAD -> master) A third commit
bbb2222 A second commit
aaa1111 A first commit
9999999 Initial commit
```

Git rebase

```
$ git log --oneline --decorate
ccc3333 (HEAD -> master) A third commit
bbb2222 A second commit
aaa1111 A first commit
9999999 Initial commit
```

--autosquash rebase

```
$ git log --oneline --decorate
ccc3333 (HEAD -> master) A third commit
bbb2222 A second commit
aaa1111 A first commit
9999999 Initial commit
```

Git commit --fixup commit --fixup

```
$ git log --oneline --decorate
ccc3333 (HEAD -> master) A third commit
bbb2222 A second commit
aaa1111 A first commit
9999999 Initial commit
```

git rebase --autosquash

```
$ git log --oneline --decorate
ccc3333 (HEAD -> master) A third commit
bbb2222 A second commit
aaa1111 A first commit
9999999 Initial commit
```

git merge --squash git merge --squash

```
git merge --squash <branch>
git commit
```

git reset

```
git merge --squash <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 fetch -p
```

。

"

```
git fetch -p
```

```
'git branch -vv' '-d'
```

```
git fetch -p
```

<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 dev
```

v.01

```
git archive --output=archive-dev.zip --prefix=src-directory-name dev
```

HEAD sub-dir

```
git archive --output=archive-dev.zip --prefix=src-directory-name dev
```

git

git archive

HEAD tar

```
git archive --format tar HEAD | cat > archive-HEAD.tar
```

gzip HEAD tar

```
git archive --format tar HEAD | cat > archive-HEAD.tar
```

tar.gz

```
git archive --format tar HEAD | cat > archive-HEAD.tar
```

HEAD zip

```
git archive --format tar HEAD | cat > archive-HEAD.tar
```

```
git archive --format tar HEAD | cat > archive-HEAD.tar
```

<https://riptutorial.com/zh-TW/git/topic/2815/>

- `git blame [ ]`
- `git blame [-f] [ - e] [ - w] [filename]`
- `git blame [-L range] [ ]`

| -F      |                                          |
|---------|------------------------------------------|
| -e      |                                          |
| -w      |                                          |
| -L      | <code>git blame -L 1,2 [filename]</code> |
| --show- | <code>blame</code>                       |
| -l      |                                          |
| -t      |                                          |
| -       |                                          |
| -p -    |                                          |
| -M      |                                          |
| -C      | <code>-M</code>                          |
| -H      |                                          |
| -C      | <code>git-annotate</code>                |
| -n      |                                          |

git blame

### Examples

`git blame <file>`

#####

在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
```

。 。 。onlineing 。 q。

```
git log -  SHA-1
```

Free Code Camp

```
git log
```

n。 2

```
git log
```

## Online

```
git log --online
```

。online。

oneline 顯示所有 commit 的摘要。 - 顯示

顯示 [Free Code Camp](#) 的 commit 歷史。

```
git log --oneline
```

顯示 commit 歷史，並顯示 commit 的摘要。 顯示 commit 歷史，並顯示 commit 的摘要。

```
git log --oneline
```

顯示 commit 歷史，並顯示 commit 的摘要。

顯示 commit 歷史，並顯示 commit 的摘要。

```
git log --decorate --oneline --graph
```

顯示 commit 歷史，並顯示 commit 的摘要。

```
git log --decorate --oneline --graph
```

顯示 commit 歷史，並顯示 commit 的摘要。

```
git log --decorate --oneline --graph
```

顯示 commit 歷史，並顯示 commit 的摘要。

```
git log --decorate --oneline --graph
```

顯示 commit 歷史，並顯示 commit 的摘要。

顯示 commit 歷史，並顯示 commit 的摘要。

```
git log --patch
```

顯示 commit 歷史，並顯示 commit 的摘要。

```
git log --patch
```

顯示 commit 歷史，並顯示 commit 的摘要。

```
git log -S"#define SAMPLES"
```

顯示 commit 歷史，並顯示 commit 的摘要。 顯示 commit 歷史，並顯示 commit 的摘要。

```
git log -S"#define SAMPLES"
```

顯示 commit 歷史，並顯示 commit 的摘要。

```
git log -S"#define SAMPLES"
```

```
git log -S"#define SAMPLES"
```

顯示 commit 歷史，並顯示 commit 的摘要。 顯示 commit 歷史，並顯示 commit 的摘要。

```
git log -S"#define SAMPLES"
```

□ □ □ □ □ □ □ □ □ □ □ □ □ □

```
git shortlog >>>>> git log >>
```

[illegible]

```
$ 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
Committer 1 (<number_of_commits>):
    Commit Message 1
    Commit Message 2
    ...
Committer 2 (<number_of_commits>):
    Commit Message 1
    Commit Message 2
    ...
```

[illegible]
$$-n$$

```
--numbered
```

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

-e

```
--email
```

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

```
--format
```

```
git log --format %H %s %e %u %d %p %C
```

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ ☒ ☒

1111

```
git log --after '3 days ago'
```



```
git log --after '3 days ago'
```

author:

```
git log --after '3 days ago'
```

```
$ git log -L 1,20:index.html
commit 6a57fde739de66293231f6204cbd8b2fec3a869
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'
```

| 変数             | 形式        |
|----------------|-----------|
| %C(color_name) | カラー名      |
| %hHH           | 16進カラーコード |
| %Creset        | カラーリセット   |
| %d             | 数値        |
| %s             | 文字列 [文字列] |
| %cr            | 改行コード     |
| %an            | 改行コード     |

11/11/2019

```
tree = log --oneline --decorate --source --pretty=format:'"%Cblue %h %Cgreen %ar %Cblue %an
%C(yellow) %d %Creset %s"' --all --graph
```

```
tree = log --oneline --decorate --source --pretty=format: '%Cblue %h %Cgreen %ar %Cblue %an
%C(yellow) %d %Creset %s"' --all --graph
```

# Git

```
git log master..foo[] foo[][][]master[]。 [] [] [] [] [] [] [] [] [] [] [] [] [] []
```

1000000000

```
git log --stat
```

□ □

```
git log --stat
```

□ □ □ □ □ □ □ □ □

```
git show
```

```
git show 48c83b3
git show 48c83b3690dfc7b0e622fd220f8f37c26a77c934
```

1

```
git show 48c83b3
git show 48c83b3690dfc7b0e622fd220f8f37c26a77c934
```

```
git log
```

```
0000000000000000git000
```

```
git log [options] --grep "search_string"
```

11

```
git log [options] --grep "search_string"
```

```

##### removed file###.

```

```
git --invert-grep --invert-grep
```

11

```
git log [options] --grep "search_string"
```

```
#####add file.
```

🔗 <https://riptutorial.com/zh-TW/git/topic/240>

54: 00

11

- `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>`

11

```

◦ <stash>◦
◦ @ {0}@ {1}◦
◦
pop◦
◦ ◦
◦ <stash><stash>◦ stash @ {0}<stash>stash @ {<revision>}◦
ref◦ ◦ ;“”◦
stash refgit stash createreflog◦ ◦ ;“”◦

```

11

[illegible]

## Examples

### Stashing

[illegible]

```
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 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 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 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 apply

```
(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 apply

```
(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
```

#####--include-untracked-u。

```
git stash
```

#####

```
git stash
```

#####--keep-index-k。

```
git stash
```

#####

```
git stash list
```

#####。  
#####

```
git stash list
```

#####stash@{1}。

#####

#####

```
git stash show
```

#####

```
git stash show
```

#####

```
git stash show
```

#####

git stash clear

```
git stash clear
```

git stash clear

```
git stash clear
```

git stash clear

```
git stash clear
```

git stash pop

git stash pop

```
git stash pop
```

git stash pop

```
git stash pop
```

git stash apply

git stash apply

```
git stash apply
```

git stash apply

```
git stash apply
```

git stash apply

git stash apply

```
git stash apply
```

git stash apply

```
git stash apply
```

git stash apply

```
git stash apply
```

git stash apply

```
git stash apply
```

git stash apply

git stash apply

```
git stash -p
```

成功儲存。。

2.13.0 版本在 `push` 時會自動執行 `paths` 檢查。

```
git stash -p
```

成功儲存。

成功儲存。

```
git checkout stash@{0} -- myfile.txt
```

成功。

Stashing 的缺點 - 儲存時會自動執行 `paths` 檢查。

成功。

成功儲存。

```
git stash --keep-index
```

成功儲存。

成功。

Stash 的缺點 - 儲存時會自動執行 `paths` 檢查。

```
git stash --keep-index
```

成功儲存。

成功。

成功儲存。

```
git stash --keep-index
```

Git 的缺點 - 儲存時會自動執行 `paths` 檢查。

成功。

成功儲存。

```
git stash
git checkout correct-branch
git stash pop
```

成功 `git stash pop` 成功儲存。

```
git stash
git checkout correct-branch
git stash pop
```



```
git stash pop
```

```
$ git stash pop
[...]
Dropped refs/stash@{0} (2ca03e22256be97f9e40f08e6d6773c7d41dbfd1)
```

```
git stash drop
```

1111111111111111

```
$ git stash pop
[...]
Dropped refs/stash@{0} (2ca03e22256be97f9e40f08e6d6773c7d41dbfd1)
```

[illegible]

```

00000000000000000000000000000000 gitk 0

```

```
$ git stash pop
[...]
Dropped refs/stash@{0} (2ca03e22256be97f9e40f08e6d6773c7d41dbfd1)
```

□□□□□□□□□□□□□□□□□□ □□□□□□□□。

```
git log --graph --oneline --decorate | gitk --gui
```

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

```

somebranch  WIP  commithash

```

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

```
$ git stash pop
[...]
Dropped refs/stash@{0} (2ca03e22256be97f9e40f08e6d6773c7d41dbfd1)
```

[illegible]

🔗 <https://riptutorial.com/zh-TW/git/topic/1440>

55:

- `git show [options] <object> ...`

Git。

- 
- 

## Examples

`git show`Git。

。

| <code>git show</code>     |  |
|---------------------------|--|
| <code>git show @~3</code> |  |

blob。

| <code>git show @~3:</code>               | <b>3</b>                                 |
|------------------------------------------|------------------------------------------|
| <code>git show @~3:src/program.js</code> | <code>src/program.js</code> <b>3blob</b> |
| <code>git show @:a.txt @:b.txt</code>    | <code>b.txta.txt</code>                  |

。

<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`
  - `<git>\mingw64\etc\gitconfig` Windows
  - `$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 git config -l
" " .
```

git 2.8

```
git config --list --show-origin
```

。 。

- `core.editor`。

```
$ git config --global core.editor nano
```

- `GIT_EDITOR`。

。

```
$ git config --global core.editor nano
```

。 。

```
$ git config --global core.editor nano
```

- 的 `Git` `VISUAL_EDITOR`。 `VISUAL_EDITOR`。

```
$ git config --global core.editor nano
```

;`shell`。 `bash` `~/.bashrc` `~/.bash_profile`。

`GUI`。 `Git` `Aborting commit due to empty commit message.`。 `--wait`。

。

。

。

## Windows

Microsoft Windows `OS` - `CR + LF`。 `Unix` `Linux` `OSX` `LF`。

。

```
git config --global core.autocrlf=true
```

Microsoft Windows `LF` - `> CR + LF`。

## Unix / Linux / OSX

Unix `Microsoft Windows`。

```
git config --global core.autocrlf=true
```

`CR + LF` - `> + LF`。

。

`-c <name>=<value>`。

gitconfig

```
git -c user.email = mail@example.com commit -m "some message"
```

user.name user.email git

git

```
git config --global http.proxy http://my.proxy.com:portnumber
```

```
git config --global http.proxy http://my.proxy.com:portnumber
```

```
git config --global help.autocorrect 17
```

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 --list
...
core.editor=vim
credential.helper=osxkeychain
...
```

true --global

```
$ git config --list
...
core.editor=vim
credential.helper=osxkeychain
...
```

Git 2.13 版本開始支援此功能。

## Windows

### .gitconfig

執行 `git config --global -e`

編輯

```
[includeIf "gitdir:D:/work"]
    path = .gitconfig-work.config

[includeIf "gitdir:D:/opensource/"]
    path = .gitconfig-opensource.config
```

編輯

- 新增或修改 `"gitdir:D:/work"`。
- `gitdir:D:/work` - 指定 `"gitdir:D:/work"` 目錄。
- `gitdir:D:/opensource`。

### .gitconfig-work.config

編輯 `.gitconfig` 檔案

```
[includeIf "gitdir:D:/work"]
    path = .gitconfig-work.config

[includeIf "gitdir:D:/opensource/"]
    path = .gitconfig-opensource.config
```

### .gitconfig-opensource.config

編輯 `.gitconfig` 檔案

```
[includeIf "gitdir:D:/work"]
    path = .gitconfig-work.config

[includeIf "gitdir:D:/opensource/"]
    path = .gitconfig-opensource.config
```

## Linux

```
[includeIf "gitdir:D:/work"]
    path = .gitconfig-work.config

[includeIf "gitdir:D:/opensource/"]
    path = .gitconfig-opensource.config
```

Windows 版本支援此功能。

更多資訊請參考 <https://riptutorial.com/zh-TW/git/topic/397>

57:

## Examples

`git merge` " ".

`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

```
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 add index.html`  
`git commit`.

<https://riptutorial.com/zh-TW/git/topic/3233/>

## Examples

### Atlassian SVN to Git

Atlassian 提供了 Java 运行环境 [JRE](#)。

```
java -jar svn-migration-scripts.jar verify
git-svn
```

authors。 Subversion 到 Git 的迁移脚本。 迁移脚本将 Subversion 的 authors 文件迁移到 Git 的 authors 文件中。

```
java -jar svn-migration-scripts.jar authors <svn-repo> authors.txt
```

<svn-repo> 是 Subversion 的 URL。 迁移脚本将 authors.txt 文件迁移到 <username>@mycompany.com 的 authors 文件中。 迁移脚本将 Subversion 的 authors 文件迁移到 Git 的 authors 文件中。

svn repo 到 Git

```
java -jar svn-migration-scripts.jar authors <svn-repo> authors.txt
```

<svn-repo> 是 Subversion 的 URL。 <git-repo-name> 是 Git 仓库的名称。 迁移脚本将 Subversion 的 authors 文件迁移到 Git 的 authors 文件中。

- --stdlayout 选项将 Git 仓库的 trunk 分支迁移到 Subversion 的 trunk 分支。 迁移脚本将 Subversion 的 trunk 分支迁移到 Git 的 trunk 分支。 迁移脚本将 Subversion 的 tags 分支迁移到 Git 的 tags 分支。 迁移脚本将 Subversion 的 branches 分支迁移到 Git 的 branches 分支。 迁移脚本将 Subversion 的 bugfixes 分支迁移到 Git 的 bugfixes 分支。 迁移脚本将 Subversion 的 authors 文件迁移到 Git 的 authors 文件中。
- 迁移脚本将 Subversion 的 authors 文件迁移到 Git 的 authors 文件中。
- 迁移脚本将 Subversion 的 authors 文件迁移到 Git 的 authors 文件中。

git svn clone 迁移 Subversion 的 trunk 分支到 Git 的 trunk 分支。 迁移脚本将 Subversion 的 tags 分支迁移到 Git 的 tags 分支。 迁移脚本将 Subversion 的 branches 分支迁移到 Git 的 branches 分支。 迁移脚本将 Subversion 的 bugfixes 分支迁移到 Git 的 bugfixes 分支。 迁移脚本将 Subversion 的 authors 文件迁移到 Git 的 authors 文件中。

```
java -jar svn-migration-scripts.jar authors <svn-repo> authors.txt
```

svn 到 Git 的迁移脚本。 迁移脚本将 Subversion 的 authors 文件迁移到 Git 的 authors 文件中。

### SubGit

SubGit 提供了 SVN 到 Git 的迁移脚本。

```
$ subgit import --non-interactive --svn-url http://svn.my.co/repos/myproject myproject.git
```

### svn2git SVN 到 Git

svn2git 提供了 Ruby 实现的 git-svn 迁移脚本。 迁移脚本将 Subversion 的 trunk 分支迁移到 Git 的 trunk 分支。 迁移脚本将 Subversion 的 tags 分支迁移到 Git 的 tags 分支。 迁移脚本将 Subversion 的 branches 分支迁移到 Git 的 branches 分支。 迁移脚本将 Subversion 的 bugfixes 分支迁移到 Git 的 bugfixes 分支。 迁移脚本将 Subversion 的 authors 文件迁移到 Git 的 authors 文件中。

迁移脚本将 Subversion 的 authors 文件迁移到 Git 的 authors 文件中。

迁移脚本将 Subversion 的 authors 文件迁移到 Git 的 authors 文件中。

```
$ svn2git http://svn.example.com/path/to/repo
```



要將 SVN 儲存庫轉換為 Git 儲存庫

```
$ svn2git http://svn.example.com/path/to/repo
```

若要指定 SVN 儲存庫的 URL，請使用 `--notrunk`、`--nobranches` 或 `--notags`。

例如，若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

```
$ svn2git http://svn.example.com/path/to/repo
```

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

```
$ svn2git http://svn.example.com/path/to/repo
```

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

**Team Foundation** 與 **TFVC** 與 **Git**

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

**Git-TF**

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

```
git-tf clone http://my.tfs.server.address:port/tfs/mycollection  
'$/myproject/mybranch/mysolution' --deep
```

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

- 若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。
- 若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

```
git-tf clone http://my.tfs.server.address:port/tfs/mycollection  
'$/myproject/mybranch/mysolution' --deep
```

**Mercurial** 與 **Git**

若要指定 SVN 儲存庫的 URL，請使用 `--trunk trunk-dir --notags --nobranches`。

## 1. 安裝 Git

```
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) [https](https://stackoverflow.com/a/31827990/5283213) [/stackoverflow.com/a/31827990/5283213](https://stackoverflow.com/a/31827990/5283213)

3. [GitHub](#) [GitHub](#) [GitHub](#) .

[riptutorial.com/zh-TW/git/topic/3026/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 directoryToFolder/newName
```

```
git branch -m old_name new_name
```

```
git checkout old_branch
```

```
git checkout old_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. 100

git checkout

commit-msg git commit

pre-receive

<old-value> <new-value> <ref-name>

git commit

-MSG

pre-commit

- 
- - -m -F
  - -t
  - 
  -
- SHA1 -c -C --amend

pre-commit

git rebase

2

- 1.
- 2.

rebase

git push git push

[illegible]

```

0000000000000000ref0000000000000000000000000000

```

```
<old-value> <new-value> <ref-name>
```

11

[illegible]

00000000 30000

- `ref`
- `ref`
- `ref`

```

#####pre-receive#####ref#####update #####.

```

11

Git 1.8.2。

1.8

[illegible]

```

0000 .git/hooks/00 gotcha alert 000000pre-push0000000pre-push .git/hooks/ 0000000000 chmod +x
./git/hooks/pre-push。

```

0000.000 Hannah Wolfe0000000000000000

```
#!/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

```
#!/bin/bash

protected_branch='master'
current_branch=$(git symbolic-ref HEAD | sed -e 's,.*\/\ (.*)\,,\1,')

if [ $protected_branch = $current_branch ]
then
```

```
exit 0
git push...
git push...
```

```

[[[ https://git-scm.com/docs/githooks#_pre_push
[[[[[ https://
//github.com/git/git/blob/87c86dd14abe8db7d00b0df5661ef8cf147a72a3/templates/hooks--pre-
push.sample

```

```
.git/hooks/pre-commit
```

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

```
$ cat .git/hooks/post-receive
```

```
##### egrep regexp#####JIRA-12345#####Jira###. #####.
```

<https://riptutorial.com/zh-TW/home>

61:

- `git rev-list [options] <commit> ...`



## Examples

List master origin / master

```
git rev-list --oneline master ^origin/master
```

Git rev-list 列出所有提交。 列出所有提交。

- `--oneline` 选项。
- `^` 选项。
- 列出所有提交。 运行 `git rev-list foo bar ^baz foo bar baz`。

更多请参见 <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, obl, own sourcing dev training, Pod, Prince J, RamenChef, Rick, Roald Nefs, ronnyfm, Sazzad Hissain Khan, Scott Weldon, Sibi Raj, TheDarkKnight, theheadofabroom, ʘolęęz ęqʘ 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, lxr, penguincoder, RamenChef, SashaZd, Tyler Hyndman, vkluge                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 20 | Reflog - git   | Braiam, Peter Amidon, Scott Weldon                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 21 | TortoiseGit    | Julian, Matas Vaitkevicius                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 22 | Worktrees      | andipla, Configure, 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, mystarocks, 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 |                | AesSedail01, Andy Hayden, Asaph, Configure, intboolstring, Jakub Narebski, jkdev, Muhammad Abdullah, Nathan Arthur, ownsourcing dev training, Richard Dally, Wolfgang                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 30 |                | Boggin, Configure, Daniel Käfer, Dimitrios Mistriotis, forresthopkinsa, hardmoorth, 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, PRIYANSHU, 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 |                | AesSedail01, Ajedi32, Andy, Anthony Staunton, Asenar, bstpierre, erewok, eush77, fracz, Gaelan, jrf, jtbandes, madhead, Michael Deardeuff, mickeyandkaka, nus, penguincoder, riyadhalmur, 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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

|    |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                   | , Configure, Daniel Käfer, Derek Liu, Dunno, e.doroskevic, Enrico Campidoglio, eskwayrd, 000000, 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 | 000000differtools | AesSedail01, Micha Wiedenmann                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 37 | 000               | 321hendrik, Chin Huang, ComicSansMS, foraidt, intboolstring, J F, kowsky, mpromonet, PaladiN, tinlyx, Undo, VonC                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 38 | 00                | 4444, Jeff Puckett                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 39 | 00/0000000        | 4444, Hannoun Yassir, jornh, Kissaki, MrTux, Scott Weldon, Simone Carletti, zebediah49                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 40 | 0000000000        | AER, AesSedail01, agilob, Alex, Amitay Stern, AnimiVulpis, Ates Goral, Aukhan, Avamander, Ben, bpoiss, Braiam, bwegs, Cache Staheli, Collin M, Community, Dartmouth, David Grayson, Devesh Saini, Dheeraaj 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, nouqAdAzerO, Nuri Tasdemir, Ortomala Lokni, PaladiN, Panda, pecil, pktangyue, poke, pylang, RhysO, Rick, rokonoid, Sascha, Scott Weldon, Sebastianb, SeeuD1, sjas, Slayther, SnoringFrog, spikeheap, theJollySin, Toby, ʎoleez əqʎ qoq, Tom Gijsselinck, Tomasz Bąk, Vi., Victor Schröder, VonC, Wilfred Hughes, Wolfgang, ydaetskcoR, Yosvel Quintero, Yury Fedorov, Zaz, Zeeker |
| 41 | 00                | Creative John, Hardik Kanjariya ツ, Julie David, kisanme, 0ANAYI0 TIS, Scott Weldon, strangeqargo, Zaz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 42 | 0                 | Kissaki, MayeulC, mpromonet, rene, Ryan, Scott Weldon, Shog9, Stony, Thamilan, Thomas Gerot, Zaz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 43 | 00                | jwd630                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 44 | 0000              | Atul Khanduri, Braiam, bud-e, dubek, Florian Hämmerle, intboolstring, Julian, kisanme, Lochlan, mpromonet, RedGreenCode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 45 | 00                | AER, Cody Guldner, cringe, frlan, Guillaume, intboolstring, Mário Meyrelles, Marvin, Matt S, MayeulC, pcm, pogosama, Thomas Gerot, Tomás Cañibano                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 46 | 00                | 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 | 00             | Adi Lester, AesSedail01, 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 | 00             | adarsh, ams, AndiDog, bandi, Braiam, Caleb Brinkman, eush77, georgebrock, jpkrohling, Julian, Mateusz Piotrowski, Ortomala Lokni, RamenChef, Tall Sam, WMios                                                                                                                                                                                             |
| 49 | 00000000000000 | Thomas Crowley                                                                                                                                                                                                                                                                                                                                           |
| 50 | 00git000000    | xiaoyaoworm                                                                                                                                                                                                                                                                                                                                              |
| 51 | 00             | Dartmouth, forevergenin, Neto Buenrostro, RamenChef                                                                                                                                                                                                                                                                                                      |
| 52 | 00             | fracz, Matthew Hallatt, nighthawk454, Priyanshu Shekhar, WPrecht                                                                                                                                                                                                                                                                                         |
| 53 | 0000000        | Ahmed Metwally, Andy Hayden, Aratz, Atif Hussain, Boggin, Brett, Configure, 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 | 00             | aavrug, AesSedail01, Asaph, Brian Hinchey, bud-e, Cache Staheli, Deep, e.doroskevic, fracz, GingerPlusPlus, Guillaume, inkista, Jakub Narebski, Jared, jeffdill12, joeytwiddle, Julie David, Kara, Koraktor, Majid, manasouza, Ortomala Lokni, Patrick, Peter Mitrano, Ralf Rafael Frix, Sebastianb, Tomás Cañibano, Wojciech Kazior                     |
| 55 | 00             | Zaz                                                                                                                                                                                                                                                                                                                                                      |
| 56 | 00             | 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 | 0000000        | Braiam, Dartmouth, David Ben Knoble, Fabio, nus, Vivin George, Yury Fedorov                                                                                                                                                                                                                                                                              |
| 58 | 000Git         | AesSedail01, Boggin, Configure, Guillaume Pascal, Indregard, Rick, TheDarkKnight                                                                                                                                                                                                                                                                         |
| 59 | 000            | bud-e, Karan Desai, P.J.Meisch, PhotometricStereo                                                                                                                                                                                                                                                                                                        |
| 60 | 0              | AesSedail01, AnoE, Christiaan Maks, Configure, Eidolon, Flows, fracz, kaartic, lostphilosopher, mwarsco                                                                                                                                                                                                                                                  |
| 61 | 000            | mkasberg                                                                                                                                                                                                                                                                                                                                                 |