从当前目录初始化

```
$ git init

$ git add *.c

$ git commit -m 'initial project version'
```

从现有仓库克隆

```
$ git clone git://.....
```

检查当前文件状态

```
$ git status
```

跟踪新文件

```
$ git add README
```

查看已暂存和未暂存的更新

```
$ git diff
```

提交更新

```
$ git commit -m "Story 182: Fix benchmarks for
speed"

$ git commit -a -m 'added new benchmarks'
```

移除文件

```
$ rm grit.gemspec
$ git status
# On branch master
#
# Changed but not updated:
# (use "git add/rm <file>..." to update what will
be committed)
#
# deleted: grit.gemspec
#

$ git rm grit.gemspec
rm 'grit.gemspec'
$ git status
# On branch master
#
# Changes to be committed:
# (use "git reset HEAD <file>..." to unstage)
#
# deleted: grit.gemspec
#

$ git rm --cached readme.txt

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

移动文件

```
$ git mv README.txt README
$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 1
commit.
#
# Changes to be committed:
# (use "git reset HEAD <file>..." to unstage)
#
# renamed: README.txt -> README
#

$ mv README.txt README
$ git rm README.txt
$ git add README
```

查看提交历史

```
$ git log

最近两次
$ git log -p -2

统计
$ git log --stat

一行
$ git log --pretty=oneline

自定义格式
$ git log --pretty=format:"%h - %an, %ar : %s"

选项说明
%H 提交对象（commit）的完整哈希字串
%h 提交对象的简短哈希字串
%T 树对象（tree）的完整哈希字串
%t 树对象的简短哈希字串
%P 父对象（parent）的完整哈希字串
%p 父对象的简短哈希字串

分支
$ git log --pretty=format:"%h %s" --graph
```

修改最后一次提交

```
$ git commit --amend

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

取消已经暂存的文件

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

```
#

$ git reset HEAD benchmarks.rb
benchmarks.rb: locally modified
$ git status
# On branch master
# Changes to be committed:
# (use "git reset HEAD <file>..." to unstage)
#
# modified: README.txt
#
# Changed but not updated:
# (use "git add <file>..." to update what will be
committed)
# (use "git checkout -- <file>..." to discard
changes in working directory)
#
# modified: benchmarks.rb
#
```

取消对文件修改

```
# Changed but not updated:
# (use "git add <file>..." to update what will be
committed)
# (use "git checkout -- <file>..." to discard
changes in working directory)
#
# modified: benchmarks.rb
#

$ git checkout -- benchmarks.rb
$ git status
# On branch master
# Changes to be committed:
# (use "git reset HEAD <file>..." to unstage)
#
# modified: README.txt
#
```

查看当前远程仓库

```
$ git clone git://github.com/schacon/ticgit.git
Initialized empty Git repository in
/private/tmp/ticgit/.git/
remote: Counting objects: 595, done.
remote: Compressing objects: 100% (269/269), done.
```

```
remote: Total 595 (delta 255), reused 589 (delta
253)
Receiving objects: 100% (595/595), 73.31 KiB | 1
KiB/s, done.
Resolving deltas: 100% (255/255), done.
$ cd ticgit
$ git remote
origin

$ git remote -v
origin git://github.com/schacon/ticgit.git
```

添加远程仓库

```
$ git remote
origin
$ git remote add pb
git://github.com/paulboone/ticgit.git
$ git remote -v
origin git://github.com/schacon/ticgit.git
pb git://github.com/paulboone/ticgit.git
```

从远处仓库抓取数据

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

推送数据到远程仓库

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

查看远程仓库信息

```
$ git remote show origin
* remote origin
URL: git://github.com/schacon/ticgit.git
Remote branch merged with 'git pull' while on
branch master
master
Tracked remote branches
master
ticgit
```

远程仓库的删除和重命名

```
$ git remote rename pb paul
$ git remote
origin
paul

$ git remote rm paul
$ git remote
origin
```

列显已有的标签

```
$ git tag
v0.1
v1.3
```

含附注的标签

```
$ git tag -a v1.4 -m 'my version 1.4'
$ git tag
v0.1
v1.3
v1.4


$ git show v1.4
tag v1.4
Tagger: Scott Chacon <schacon@gee-mail.com>
Date: Mon Feb 9 14:45:11 2009 -0800
my version 1.4
commit 15027957951b64cf874c3557a0f3547bd83b3ff6
Merge: 4a447f7... a6b4c97...
Author: Scott Chacon <schacon@gee-mail.com>
Date: Sun Feb 8 19:02:46 2009 -0800
Merge branch 'experiment'
```

签署标签

```
$ git tag -s v1.5 -m 'my signed 1.5 tag'
```

```
You need a passphrase to unlock the secret key for
user: "Scott Chacon <schacon@gee-mail.com>"
1024-bit DSA key, ID F721C45A, created 2009-02-09
```

轻量级标签

```
$ git tag v1.4-lw
```

后期加注标签

```
$ git log --pretty=oneline
15027957951b64cf874c3557a0f3547bd83b3ff6 Merge
branch 'experiment'
a6b4c97498bd301d84096da251c98a07c7723e65 beginning
write support
0d52aaab4479697da7686c15f77a3d64d9165190 one more
thing
6d52a271eda8725415634dd79daabbc4d9b6008e Merge
branch 'experiment'
0b7434d86859cc7b8c3d5e1dddfed66ff742fcbc added a
commit function
4682c3261057305bdd616e23b64b0857d832627b added a
todo file

$ git tag -a v1.2 9fceb02
```

分享标签

```
$ git push origin v1.5
Counting objects: 50, done.
Compressing objects: 100% (38/38), done.
Writing objects: 100% (44/44), 4.56 KiB, done.
Total 44 (delta 18), reused 8 (delta 1)
To git@github.com:schacon/simplegit.git
* [new tag] v1.5 -> v1.5

$ git push origin --tags
```

分支

新建分支

```
$ git branch [branchname]
```

切换到其他分支

```
$ git checkout [branchname]
```

新建并且还到分支

```
$ git checkout -b [branchname]
```

创建紧急修补分支，合并到主分支

```
$ git checkout -b 'hotfix'

$ git checkout master
$ git merge hotfix
```

Fast forward

请注意，合并时出现了“Fast forward”（快进）提示。由于当前 master 分支所在的
commit 是要并入的hotfix 分支的直接上游，Git 只需把指针直接右移。换句话说，如果
顺着一个分支走下去可以到达另一个分支，那么Git 在合并两者时，只会简单地把指针前
移，因为没有什么分歧需要解决，所以这个过程叫做快进（Fast forward）。

删除分支

```
$ git branch -d [branchname]
```

基本合并

```
$ git checkout master
$ git merge iss53
Merge made by recursive.
README | 1 +
```

```
1 files changed, 1 insertions(+), 0 deletions(-)
```

冲突合并

远程分支

```
(远程仓库名)/(分支名)
```

检出远程分支

```
$ git checkout --track -b xsf_1.9_LTS
origin/xsf_1.9_LTS
```

推送

```
$ git push origin [localbranchname]
```

跟踪分支

```
。在跟踪分支里输入git push，Git 会自行推断应该向哪个服
务器的哪个分支推送数据。反过来，在这些分支里运行git pull 会获
取所有远程索引，并
把它们的数据都合并到本地分支中来。
在克隆仓库时，Git 通常会自动创建一个master 分支来跟踪
origin/master。这正是
git push 和git pull 一开始就能正常工作的原因。
```

```
$ git checkout --track origin/serverfix
Branch serverfix set up to track remote branch
refs/remotes/origin/serverfix.
Switched to a new branch "serverfix"

$ git checkout -b sf origin/serverfix
Branch sf set up to track remote branch
refs/remotes/origin/serverfix.
Switched to a new branch "sf"
```

删除远程分支

```
$ git push origin :serverfix
To git@github.com:schacon/simplegit.git
- [deleted] serverfix
```

update submodule

```
 git submodule init
 git submodule update
```

CMD Tips (last edited 2012-02-02 08:43:06 by XinNi)