

Git常用命令集合部曲（一）

刘宝龙 IT行业技术圈 今天



点击蓝字



进入下一步

• 代码库

- init
- clone

• 在线帮助

- help

• 日常变更

- status
- diff
- add
- commit
- stash
- rm
- mv

• 日志

- log
- diff
- show
- blame
- grep

• 分支

- branch
- checkout
- tag

• 历史反转

- reset
- revert

• 合并

- cherry-pick
- merge
- mergetool
- rebase

• 清理

- clean
- gc

 Linux微堂笔记

基于本地目录创建Git仓库 (init)

通过示例说明：

```
$ mkdir git-practice
$ cd git-practice
$ git init # 初始化git库
Initialized empty Git repository in /root/git-practice/.git/
```

- git通常作为隐藏目录存在（.git/）

```
$ ls .git/
branches  COMMIT_EDITMSG  config  description  HEAD  hooks  index  info
```

直接位于 .git/ 路径下的文件：

- .git/COMMIT_EDITMSG：最近一次运行 `git commit` 时附加的消息

- `.git/config`: Git仓库的配置文件
- `.git/description`: 当前git仓库的描述, 主要给gitweb等git托管系统使用
- `.git/HEAD`: 包含了一个分支 (branch) 的引用, 通过该文件Git可以得到下一次commit的parent
- `.git/index`: 暂存区 (stage), 是一个二进制文件
- `.git/info`: 包含仓库的一些信息;

直接位于 `.git/` 路径下的目录:

- `.git/objects`: 存放所有Git对象, 对象的SHA1哈希值的前2位是目录名, 后38位作为对象文件名;
- `.git/hooks`: 存放一些shell脚本, 可设置特定的git命令后触发相应的脚本;
- `.git/logs`: 保存所有更新的引用记录 【提交过一次版本后, 自动创建】;
- `.git/refs`: 具体的引用, 该目录包含3个子目录 (heads, remotes和tags) ;

完成一次初始提交 (add、status、diff、commit)

命令使用:

```
$ git add readme ## 添加到缓冲区【将readme文件添加到暂存区】
$ git status ## 查看当前提交状态
$ git commit -m "[提交说明]" ## 提交git【将添加到暂存区的文件提交】
$ git diff ## 比对提交的文件
```

示例说明:

```
$ echo "This is a readme file for git practice project." > readme
$ git add readme
$ git status
$ git commit -m "第一次提交"
```

```

[root@ansible-master git-test]# echo "This is a readme file for git practice project." > readme
[root@ansible-master git-test]# git status
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       readme
nothing added to commit but untracked files present (use "git add" to track)
[root@ansible-master git-test]#
[root@ansible-master git-test]# git add readme
[root@ansible-master git-test]# git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   readme

```

git add 添加到 暂存区 之前的状态为: 红色

横线部分为: 分支

git add 添加到 暂存区 之后的状态: 绿色

```

[root@ansible-master git-test]# git commit -m "第一次提交"
[master (root-commit) f63c440] 第一次提交
 1 file changed, 1 insertion(+)
 create mode 100644 readme

```

git commit 提交之后, 状态为: 无

```

[root@ansible-master git-test]# git status
# On branch master
nothing to commit, working directory clean

```

 Linux微堂笔记

图中 git status 状态信息颜色部分是开启了颜色显示:

```

# 开启颜色显示
git config --global color.ui auto

```

在进行 git commit 提交命令时, 需要配置用户信息:

```

# 配置用户信息
git config --global user.name "Your Name"
git config --global user.email "Your Email"

```

- git diff 命令, 比对文件:

```

# 查看文件在 工作目录 和 暂存区 的差别;
# 如果还没 add 进暂存区, 则查看文件自身修改前后的差别。也可查看和另一分支的区别。
git diff

```

查看日志信息并修改提交 (log、amend)

命令使用:

```

git log # 显示提交后的历史记录【它是倒序, 最新的提交处于最上面】
git log <id> --stat # 通过参数<--stat>显示修改内容
git log <id> -<number> --stat # 通过参数<number>来约束显示历史条目

```

git log HEAD -1 --stat # 通过<HEAD>参数，引用git目录HEAD文件的提交ID值（也叫哈希值）
 git log <commit id> -1 -p # -p参数表示显示修改了哪些内容等信息

amend并不是Git的命令，而是 commit命令中的一个参数

git commit --amend

git log命令的使用：

```
[root@ansible-master git-practice]# git add .
[root@ansible-master git-practice]# git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   readme
#
[root@ansible-master git-practice]# git commit -m "this is sencond to commit."
[master 1910e1e] this is sencond to commit.
1 file changed, 2 insertions(+)
[root@ansible-master git-practice]# git log
commit 1910e1ec7cc9ab97d15b4e3da3192b18345bfd1a
Author: root <liubaolong>
Date:   Wed Jul 4 22:27:02 2018 -0400

    this is sencond to commit.

commit 913e401fffb7a8fa18488861f4351230cd44ed88f
Author: root <liubaolong>
Date:   Wed Jul 4 20:41:23 2018 -0400

    initial commit
[root@ansible-master git-practice]#
```

这是40位字符的ID信息；
 它的前7位字符显示在commit提交；
 这是提交用户信息 和 提交时间
 这是commit说明信息；
 历史提交的信息【倒序】
 最新提交的信息

查看历史内容：

git log 0426396d24016a8ca94f1f64dcf5cd18b1b6d25c -1 -p

```
[root@ansible-master git-test]# git log 0426396d24016a8ca94f1f64dcf5cd18b1b6d25c -1 -p
commit 0426396d24016a8ca94f1f64dcf5cd18b1b6d25c
Author: root <liubaolong>
Date:   Thu Jul 5 02:06:17 2018 -0400

    第2次提交

diff --git a/readme b/readme
index 8331f0c..3033d66 100644
--- a/readme
+++ b/readme
@@ -1 +1,2 @@
 This is a readme file for git practice project.
+This is a readme file two.
```

显示 这次提交的信息中，
 在第几行修改了哪些内容；
 -p 参数 类似于 diff 命令

修改合并

- amend 并不是Git里面的一个命令，而是commit命令中的一个参数；
- 它主要的作用是：将修改的文件进行合并【不是合并分支哟】


```
$ echo "add one line." > makefile
```

```
$ git add makefile
```

```
$ git commit --amend # 命令提交后，进入强制编辑输入要提交的说明信息
```

```
$ git log # 进行查看历史记录，本次提交的情况
```

```
[root@ansible-master git-test]# vim Makefile
[root@ansible-master git-test]# git add Makefile
[root@ansible-master git-test]# git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   Makefile
#
# 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:   readme
#
[root@ansible-master git-test]# git commit --amend
[master 8af4dad] readme: add one line
3 files changed, 3 insertions(+)
create mode 100644 Makefile
create mode 100644 makefile
[root@ansible-master git-test]# git log
commit 8af4dada555f73ae5ebb6ed3f73e13c0ed03495b
Author: root <liubaolong>
Date: Thu Jul 5 02:06:17 2018 -0400

    readme: add one line

commit f63c440e1b0f0158f3fd8d10a08ed6c222b8e5b4
Author: root <liubaolong>
Date: Thu Jul 5 01:49:49 2018 -0400
```

第一次提交

 Linux微堂笔记

文章转载自公众号

 Linux微堂笔记 >