

[Coding Style](#)

- [社区](#)
- [公开课](#)

- [注册](#)
- [登录](#)

-

闯过这 54 关，点亮你的 Git 技能树 (二)

[Git](#) · [hkliya](#) · 于 4 月前发布 · 最后由 [hkliya](#) 于 4 月前回复 · 434 次阅读



这是一个系列文章，介绍了学习 Git 的好工具 - github，如果你是第一次看到，请先阅读：

[闯过这 54 关，点亮你的 Git 技能树](#)

[闯过这 54 关，点亮你的 Git 技能树 \(一\)](#)

今天我将带大家完成第 11 - 20 关。如对任何命令有疑问请看第一篇里的推荐教程。

第十一关

```
*****
*                               Github                               *
*****
How many changes are going to be committed? 2
Congratulations, you have solved the level!

Name: rm
Level: 11
Difficulty: **

A file has been removed from the working tree, however the file was not removed from the repository. Find out what this file was and remove it.
```

在 git 里，创建一个新文件时，要用 git add 来告诉 git，同样，删除一个文件时候，也要告诉 git。

```
→ git_hug git:(master) X gst
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    deleteme.rb

no changes added to commit (use "git add" and/or "git commit -a")
→ git_hug git:(master) X git rm deleteme.rb
rm 'deleteme.rb'
→ git_hug git:(master) X github
*****
*                               Github                               *
*****
Congratulations, you have solved the level!

Name: rm_cached
Level: 12
Difficulty: **

A file has accidentally been added to your staging area, find out which file and remove it from the staging area. *NOTE* Do not remove the file from th
```

第十二关

有时候执行 add 操作的时候不小心把多余的文件 add 进去了，这时我们需要把它从 staging area 移除出来，但不能删除文件。通过 `git help rm`，搜索 `cached`，可以看到有这个参数，刚好能满足我们的需求。

```
--cached
  Use this option to unstage and remove paths only from the index. Working tree files, whether modified or not, will be left alone.
```

```
x> ~/D/git_hug on master x gst
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   deleteme.rb

x> ~/D/git_hug on master x git rm deleteme.rb
error: the following file has changes staged in the index:
    deleteme.rb
(use --cached to keep the file, or -f to force removal)
x> ~/D/git_hug on master x git rm deleteme.rb --cached
rm 'deleteme.rb'
x> ~/D/git_hug on master x githug
*****
*                                     Github                                     *
*****
Congratulations, you have solved the level!

Name: stash
Level: 13
Difficulty: **

You've made some changes and want to work on them later. You should save them, but don't commit t
```

第十三关

这一关的场景也非常常见，回想一下以前我们用 CVS 或 SVN 的时候。

你写一个新需求写的正 High，突然你的主管找到你说有一个用户打电话来遇到一个问题，非常紧急。

我以前的做法是：把当前改过的文件「复制」一份先放在其它地方，把代码还原到没修改的状态，签出对应 production 环境的代码。定位修复问题，提交代码，确认问题修复后再合并回开发分支，并把之前备份的代码复制进来进行合并。

stash 正是解决这个问题的，只需要 `git stash`，它就把当前未提交的改动「复制」到另一个地方暂存起来，待要恢复的时候执行 `git stash pop` 即可。

```
✎> ~/D/git_hug on master ✕ githubg
*****
*                                     Github                                *
*****
Congratulations, you have solved the level!

Name: stash
Level: 13
Difficulty: **

You've made some changes and want to work on them later. You should save them, but don't commit them.

✎> ~/D/git_hug on master ✕ git stash
Saved working directory and index state WIP on master: 0206059 Add some lyrics
HEAD is now at 0206059 Add some lyrics
✎> ~/D/git_hug on master ◦ githubg
*****
*                                     Github                                *
*****
Congratulations, you have solved the level!

Name: rename
Level: 14
Difficulty: ***

We have a file called `oldfile.txt`. We want to rename it to `newfile.txt` and stage this change.
```

第十四关

这一关有两种解法，第一种是：

```
mv oldfile.txt newfile.txt
git rm oldfile.txt
git add newfile.txt
```

显然这样做用户体验很糟，Linux 肯定不会这么设计软件的。

第二种做法：

```
git mv oldfile.txt newfile.txt
```

```
«> ~/D/git_hug on master ◦ git mv oldfile.txt newfile.txt
«> ~/D/git_hug on master ◦ githubg

*****
*                                     Github                                     *
*****
Congratulations, you have solved the level!

Name: restructure
Level: 15
Difficulty: ***

You added some files to your repository, but now realize that your project needs to be restructured. Make a
` and using Git move all of the .html files into this folder.
```

第十五关

这一关相当于上一关的增强版，可以用通配符一起批量移动文件。

```
«> ~/D/git_hug on master ◦ mkdir src
«> ~/D/git_hug on master ◦ git mv *.html src/
«> ~/D/git_hug on master ◦ githubg

*****
*                                     Github                                     *
*****
Congratulations, you have solved the level!

Name: log
Level: 16
Difficulty: **

You will be asked for the hash of most recent commit. You will need to investigate the logs of the r
```

第十六关

有时候我们要看代码的提交历史，比如 [Code Review](#) 的时候。

不过 git 自带的 `git log` 命令比较弱，建议使用 GUI 客户端或 `tig`。

```
x> ~/D/git_hug on master ◦ git log
commit e3fab1127c994ab45e0587a748e01f3b26f0b7bc
Author: Seaborn Lee <seabornlee.cn@gmail.com>
Date:   Sun Jan 3 17:41:59 2016 +0800

    THIS IS THE COMMIT YOU ARE LOOKING FOR!
x> ~/D/git_hug on master ◦ githubg
*****
*                                     Github                                     *
*****
What is the hash of the most recent commit? e3fab1127c994ab45e0587a748e01f3b26f0b7bc
Congratulations, you have solved the level!

Name: tag
Level: 17
Difficulty: **

We have a git repo and we want to tag the current commit with `new_tag`.
```

第十七关

```
x> ~/D/git_hug on master ◦ git tag new_tag
x> ~/D/git_hug on master ◦ githubg
*****
*                                     Github                                     *
*****
Congratulations, you have solved the level!

Name: push_tags
Level: 18
Difficulty: **

There are tags in the repository that aren't pushed into remote repository. Push them now.

From /var/folders/x4/7l9lwjm95nv66jcfmqcvfvx80000gn/T/d20160103-19800-1w5snd6/
* [new branch]      master      -> origin/master
```

第十八关

git push 命令默认是不会 push Tags 的，需要加参数。

```
<> ~/D/git_hug on master ◦ git push --tags
Total 0 (delta 0), reused 0 (delta 0)
To /var/folders/x4/7l9lwjm95nv66jcfmqcvfvx80000gn/T/d20160103-19800-1w5snd6/.git
* [new tag]          tag_to_be_pushed -> tag_to_be_pushed
<> ~/D/git_hug on master ◦ githug

*****
*                               Github                               *
*****
Congratulations, you have solved the level!

Name: commit_amend
Level: 19
Difficulty: **

The `README` file has been committed, but it looks like the file `forgotten_file.rb` was missing from the commit.
Please amend your previous commit to include it.
```

第十九关

有时候提交之后发现漏掉了某些文件，怎么办？

往往很多人就会选择再单独提交一次，这样做其实是不合理的，之前的 commit 就不完整了，有可能上了 CI 就会挂掉。

好的做法是 amend：

```
✎> ~/D/git_hug on master ✕ git add forgotten_file.rb
✎> ~/D/git_hug on master ✕ git commit --amend
[master 7aab1f7] Initial commit
Date: Sun Jan 3 18:24:31 2016 +0800
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README
create mode 100644 forgotten_file.rb
✎> ~/D/git_hug on master ◦ githug
*****
*                                     Githug                                *
*****
Congratulations, you have solved the level!

Name: commit_in_future
Level: 20
Difficulty: **

Commit your changes with the future date (e.g. tomorrow).
```

命令执行后会调用默认的编辑器编辑 commit message，它会自动带出之前的 message，如果不需要修改，直接保存退出即可。

第二十关

默认提交的时候记录的是当前系统时间，但这一关要求覆盖提交日期。

我想不到有什么场景需要这么做，邪恶一点，有可能你把周五的提交，日期写成周末，这样表示周末在加班。

纯属YY，千万不要学，学了也不要告诉别人是我教你的。

git help commit，搜索 date:

```
--date=<date>
    Override the author date used in the commit.
```

```
<> ~/D/git_hug on master . git commit -m "I work so hard" --date "2016-01-04 23:59:59"
[master (root-commit) 36dcad5] I work so hard
Date: Mon Jan 4 23:59:59 2016 +0800
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README
<> ~/D/git_hug on master . githug
*****
*                                     *
*                               Githug                               *
*                                     *
*****
Congratulations, you have solved the level!

Name: reset
Level: 21
Difficulty: **

There are two files to be committed. The goal was to add each file as a separate commit, however both were a
stage the file `to_commit_second.rb` using the reset command (don't commit anything).
```

今天就到这里了，如果想第一时间得到更新，请关注 CodingStyle.cn !

「[软件匠艺社区](#)」旨在传播匠艺精神，通过分享好的「工作方式」和「习惯」以帮助程序员更加快乐高效地编程。

共收到 8 条回复



[horance](#) · #1 · 4 月前

第20关挺二的



[tomeokin](#) · #2 · 4 月前

第15关在win7下git2.5.0，使用通配符就会报错
执行

```
git mv *.html src/
```

```
fatal: bad source, source=*.html, destination=src/*.html
```

src是创建好的，单个文件移动就可以。



[hkliya](#) · #3 · 4 月前

#2楼 @tomeokin 加个引号呢？

```
git mv "*.html" src
```

