

Coding Style

- [社区](#)
- [公开课](#)

- [注册](#)
- [登录](#)

-

# 闯过这 54 关，点亮你的 Git 技能树 (四)

Git · [hkliya](#) · 于 2 天前发布 · 2392 次阅读



这是一个系列文章，介绍学习 Git 的一个小游戏 - githug，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 \(一\)](#)  
[闯过这 54 关，点亮你的 Git 技能树 \(二\)](#)  
[闯过这 54 关，点亮你的 Git 技能树 \(三\)](#)  
最近得了一场很严重的病差点儿没挺过来 - 懒癌。让关注此系列的朋友久等了，在这里向你致以最真诚的歉意。

今天我将带大家完成第 31 - 40 关，如对任何命令使用有疑问请看第一篇里的推荐教程。

## 第三十一关

```
*****
*                               Github                               *
*****
resetting level

Name: branch
Level: 31
Difficulty: *

You want to work on a piece of code that has the potential to break things, create the branch test_code.
```

当准备做的事情有可能会破坏其它东西时，为了不影响其他同事的开发工作，我们通常会拉一个分支出来，在分支上去做修改。

```
→ git_hug git:(master) git branch test_code
→ git_hug git:(master) gb
* master
  test_code
→ git_hug git:(master) gh
*****
*                               Github                               *
*****
Congratulations, you have solved the level!

Name: checkout
Level: 32
Difficulty: **

Create and switch to a new branch called my_branch. You will need to create a branch like you did in the previous level.
```

## 第三十二关

上一条命令只是创建了一个新的分支，并没有 checkout 过去，习惯做法通常是直接 `git checkout -b xxx`，创建并 checkout 到新的分支。如果使用 oh-my-zsh 的 git 插件的话，可以用 `gbc`，意思是：git branch create。

```
→ git_hug git:(master) git checkout -b my_branch
Switched to a new branch 'my_branch'
→ git_hug git:(my_branch) gh
*****
*                               Github                               *
*****
Congratulations, you have solved the level!

Name: checkout_tag
Level: 33
Difficulty: **

You need to fix a bug in the version 1.2 of your app. Checkout the tag `v1.2`.
```

## 第三十三关

版本 1.2 存在 bug，这里我们需要切换到 1.2 的代码以定位问题。Checkout tag 和分支没有什么区别。

```
→ git_hug git:(master) git checkout v1.2
Note: checking out 'v1.2'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

    git checkout -b <new-branch-name>

HEAD is now at 290f587... Some more changes
→ git_hug git:(290f587) githug
*****
*                               Github                               *
*****
Congratulations, you have solved the level!

Name: checkout_tag_over_branch
Level: 34
Difficulty: **

You need to fix a bug in the version 1.2 of your app. Checkout the tag `v1.2` (Note: There is also a branch named `v1.2`).
```

## 第三十四关

但当存在同名的 tag 和分支时，git 不知道我们究竟是要 checkout 到 tag 还是到分支，它认为分支的优先级更高。这时就要显式地告诉 git 我们是要切换到 tag。

```
→ git_hug git:(master) git checkout v1.2
warning: refname 'v1.2' is ambiguous.
Switched to branch 'v1.2'
→ git_hug git:(v1.2) git checkout tags/v1.2
Note: checking out 'tags/v1.2'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

    git checkout -b <new-branch-name>

HEAD is now at 6b034e7... Some more changes
→ git_hug git:(6b034e7) gh
*****
*                               Github                               *
*****
Congratulations, you have solved the level!

Name: branch_at
Level: 35
Difficulty: ***

You forgot to branch at the previous commit and made a commit on top of it. Create branch test_branch at the commit before the last.
```

## 第三十五关

有时忘记开新的分支，就修改并提交代码。开分支的时候默认是基于最新的一次提交的，但我们也可以指定参数使其基于任一次提交。

```
→ git_hug git:(master) git branch test_branch HEAD~1
→ git_hug git:(master) github
*****
*                               Github                               *
*****
Congratulations, you have solved the level!

Name: delete_branch
Level: 36
Difficulty: **

You have created too many branches for your project. There is an old branch in your repo called 'delete_me', you should delete it.
```

## 第三十六关

分支开太多就不好管理，不管使用哪种分支模型，只有很少的分支会长期存在，大部分分支都是临时的，在代码合并后就会删除掉。

```
→ git_hug git:(master) git branch -d delete_me
Deleted branch delete_me (was b60afe2).
→ git_hug git:(master) githug
*****
*                               Github                               *
*****
Congratulations, you have solved the level!

Name: push_branch
Level: 37
Difficulty: **

You've made some changes to a local branch and want to share it, but aren't yet ready to merge it with the 'master' branch. Push only 'test_branch' to
```

## 第三十七关

有时候在特性分支上提交了代码，但还不能并入主干，却又希望和别的同事分享（比如需要他们帮做 Code Review），那就需要把分支 push 到远程仓库中去。

```
→ git_hug git:(master) git push origin test_branch
Counting objects: 6, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 582 bytes | 0 bytes/s, done.
Total 6 (delta 3), reused 0 (delta 0)
To /var/folders/bm/_hw2nxj51s9c271blpgwnm7c0000gp/T/d20160514-15059-8px436/.git
 * [new branch]      test_branch -> test_branch
→ git_hug git:(master) githug
*****
*                               Github                               *
*****
Congratulations, you have solved the level!

Name: merge
Level: 38
Difficulty: **

We have a file in the branch 'feature'; Let's merge it to the master branch.
```

## 第三十八关

将另一个分支并入当前工作分支。

```
→ git_hug git:(master) git merge feature
Updating e12277f..cc8ea5a
Fast-forward
 file2 | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file2
→ git_hug git:(master) githug
*****
*                               Github                               *
*****
Congratulations, you have solved the level!

Name: fetch
Level: 39
Difficulty: **

Looks like a new branch was pushed into our remote repository. Get the changes without merging them with the local repository
```

## 第三十九关

当远程仓库有更新，但我们并不想合并到本地仓库，只想把代码拿下来看看，我们会用到 `fetch` 命令。

```
→ git_hug git:(master) git fetch origin
remote: Counting objects: 2, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (2/2), done.
From /var/folders/bm/_hw2nxj51s9c271b1pgwnm7c0000gp/T/d20160514-15150-1rnnv1qx/
 * [new branch]      new_branch -> origin/new_branch
→ git_hug git:(master) githug
*****
*                               Github                               *
*****
Congratulations, you have solved the level!

Name: rebase
Level: 40
Difficulty: **

We are using a git rebase workflow and the feature branch is ready to go into master. Let's rebase the feature branch onto our master branch.
```

## 第四十关

Rebase 这里如果不理解，请看第一篇里的推荐教程。

```
→ git_hug git:(master) git rebase master feature
First, rewinding head to replay your work on top of it...
Applying: add feature
→ git_hug git:(feature) githug
*****
*                               Github                               *
*****
Congratulations, you have solved the level!
```