

# 标签管理

发布一个版本时，我们通常先在版本库中打一个标签（tag），这样，就唯一确定了打标签时刻的版本。将来无论什么时候，取某个标签的版本，就是把那个打标签的此刻的历史版本取出来。所以，标签也是版本库的一个快照。

相当于是给我们的代码加上一个版本号，随时可以找到历史库的某个版本的完整代码，有助于我们的版本回溯

Git的标签虽然是版本库的快照，但其实它就是指向某个commit的指针（跟分支很像对不对？但是分支可以移动，标签不能移动），所以，创建和删除标签都是瞬间完成的。

Git有commit，为什么还要引入tag？

“请把上周一的那个版本打包发布，commit号是6a5819e...”

“一串乱七八糟的数字不好找！”

如果换一个办法：

“请把上周一的那个版本打包发布，版本号是v1.2”

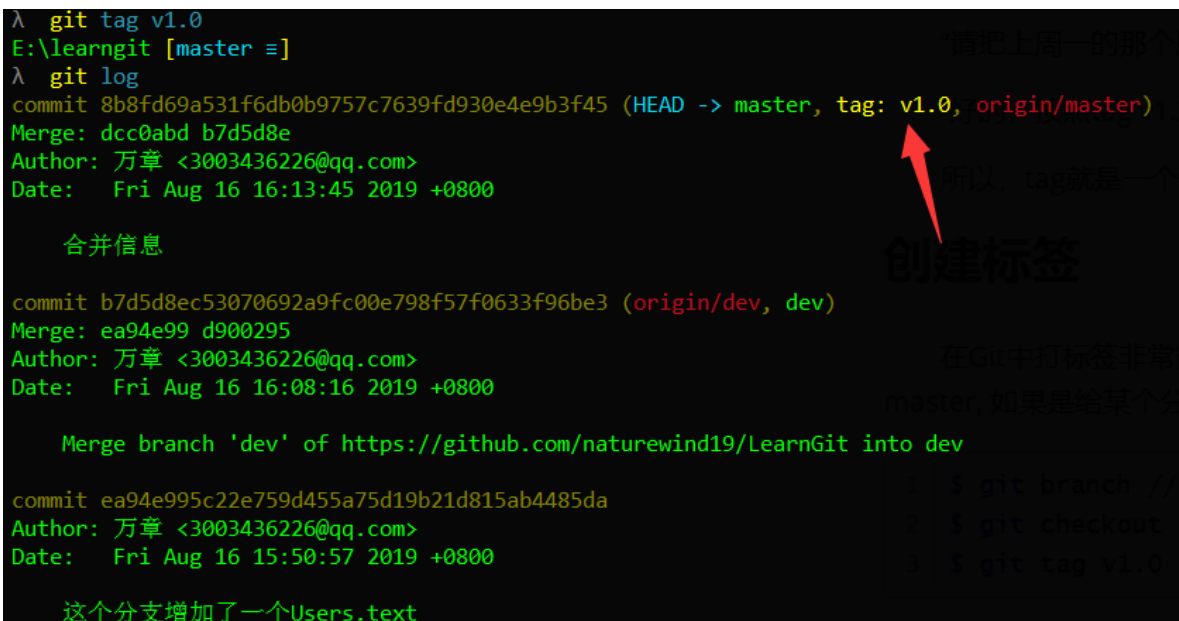
“好的，按照tag v1.2查找commit就行！”

所以，tag就是一个让人容易记住的有意义的名字，它跟某个commit绑在一起。

## 创建标签

在Git中打标签非常简单，首先，切换到需要打标签的分支上(如果是整个的大版本设置标签，一般是切换到master，如果是给某个分支设置标签，那么就是切换到目标分支)

```
1 $ git branch //查看当前分支
2 $ git checkout master //切换至主分支
3 $ git tag v1.0
```



```
λ git tag v1.0
E:\learngit [master ≡]
λ git log
commit 8b8fd69a531f6db0b9757c7639fd930e4e9b3f45 (HEAD -> master, tag: v1.0, origin/master)
Merge: dcc0abd b7d5d8e
Author: 万章 <3003436226@qq.com>
Date: Fri Aug 16 16:13:45 2019 +0800

    合并信息

commit b7d5d8ec53070692a9fc00e798f57f0633f96be3 (origin/dev, dev)
Merge: ea94e99 d900295
Author: 万章 <3003436226@qq.com>
Date: Fri Aug 16 16:08:16 2019 +0800

    Merge branch 'dev' of https://github.com/naturewind19/LearnGit into dev

commit ea94e995c22e759d455a75d19b21d815ab4485da
Author: 万章 <3003436226@qq.com>
Date: Fri Aug 16 15:50:57 2019 +0800

    这个分支增加了一个Users.txt
```

这样就给最新的一个版本打上了标签

## 给历史的某个节点补标签

比如我们的代码开发异常顺利, 顺利到, 现在已经是周五了, 但是我们忘记了在周三完成demo之后打上标签了, 此时,我们可以根据 git log 查找到历史的某个记录, 然后给它补上标签

```
λ git log
commit 8b8fd69a531f6db0b9757c7639fd930e4e9b3f45 (HEAD -> master, tag: v1.0, origin/master)
Merge: dcc0abd b7d5d8e
Author: 万章 <3003436226@qq.com>
Date: Fri Aug 16 16:13:45 2019 +0800

    合并信息

commit b7d5d8ec53070692a9fc00e798f57f0633f96be3 (origin/dev, dev)
Merge: ea94e99 d900295
Author: 万章 <3003436226@qq.com>
Date: Fri Aug 16 16:08:16 2019 +0800

    Merge branch 'dev' of https://github.com/naturewind19/LearnGit into dev

commit ea94e995c22e759d455a75d19b21d815ab4485da
Author: 万章 <3003436226@qq.com>
Date: Fri Aug 16 15:50:57 2019 +0800

    这个分支增加了一个Users.text
```

假设给该版本设置一个tag

```
1 | $ git tag v0.9 b7d5d8ec // git tag 版本号 需要打标签的那个commit的版本号
```

```
λ git log
commit 8b8fd69a531f6db0b9757c7639fd930e4e9b3f45 (HEAD -> master, tag: v1.0, origin/master)
Merge: dcc0abd b7d5d8e
Author: 万章 <3003436226@qq.com>
Date: Fri Aug 16 16:13:45 2019 +0800

    合并信息

commit b7d5d8ec53070692a9fc00e798f57f0633f96be3 (tag: v0.9, origin/dev, dev)
Merge: ea94e99 d900295
Author: 万章 <3003436226@qq.com>
Date: Fri Aug 16 16:08:16 2019 +0800

    Merge branch 'dev' of https://github.com/naturewind19/LearnGit into dev
```

```
1 | $ git tag // 该指令可以查看当前的标签列表
```

```
λ git tag
v0.9
v1.0
```

注意, 标签不是按时间顺序列出, 而是按字母排序的

```
1 | $ git show v0.9 // 查看特定版本的提交信息
```

```
λ git show v0.9
commit b7d5d8ec53070692a9fc00e798f57f0633f96be3 (tag: v0.9, origin/dev, dev)
Merge: ea94e99 d900295
Author: 万章 <3003436226@qq.com>
Date: Fri Aug 16 16:08:16 2019 +0800

    Merge branch 'dev' of https://github.com/naturewind19/LearnGit into dev
```

还可以创建带有说明的标签, 用 `-a` 指定标签名, `-m` 指定说明文字:

```
1 | $ git tag -a v0.1 -m "version 0.1 初始版" 01ec74f75
```

```
commit 34e5c8bd57a7f4afd4dee606bfe49b7bec1cd283
Author: 万章 <3003436226@qq.com>
Date:   Wed Aug 14 20:33:30 2019 +0800
```

新增一行描述信息

```
commit 01ec74f75f94a04aa22786226d916f68b3166c4d
Author: 万章 <3003436226@qq.com>
Date:   Wed Aug 14 16:59:13 2019 +0800
```

提交了一个文本信息

添加了说明信息的标签在展示信息的时候, 会展示的比较完整

```
λ git tag -a v0.1 -m "version 0.1 初始版" 01ec74f75
E:\learngit [master ≡]
λ git show v0.1
tag v0.1
Tagger: 万章 <3003436226@qq.com>
Date:   Fri Aug 16 23:26:45 2019 +0800

version 0.1 初始版

commit 01ec74f75f94a04aa22786226d916f68b3166c4d (tag: v0.1)
Author: 万章 <3003436226@qq.com>
Date:   Wed Aug 14 16:59:13 2019 +0800

提交了一个文本信息

diff --git a/readme.txt b/readme.txt
new file mode 100644
index 0000000..e627b2b
--- /dev/null
+++ b/readme.txt
@@ -0,0 +1,2 @@
+git是一个版本控制系统
+git是不用氪金的
\ No newline at end of file
```

标签总是和某个commit挂钩。如果这个commit既出现在master分支, 又出现在dev分支, 那么在这两个分支上都可以看到这个标签。

## 操作标签

###删除标签

```
1 | $ git tag -d v0.1
```

因为创建的标签都只存储在本地，不会自动推送到远程。所以，打错的标签可以在本地安全删除。

```
λ git tag
v0.9
v1.0
```

如果要推送某个标签到远程，使用命令 `git push origin <tagname>`：

```
λ git push origin v1.0
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/naturewind19/LearnGit.git
* [new tag]          v1.0 -> v1.0
```

或者，一次性推送全部尚未推送到远程的本地标签：

```
λ git push origin --tags
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/naturewind19/LearnGit.git
* [new tag]          v0.9 -> v0.9
```

如果标签已经推送到远程，要删除远程标签就麻烦一点，先从本地删除：

```
1 | $ git tag -d v0.9
```

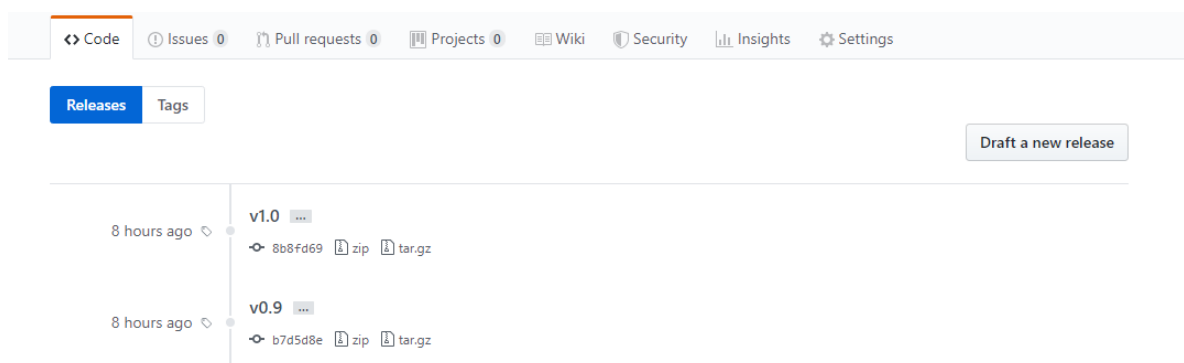
然后，从远程删除。删除命令也是push，但是格式如下：

```
1 | $ git push origin :refs/tags/v0.9
```

我们可以用版本号来进行版本回滚

```
λ git reset v0.9
E:\learngit [master |4]
λ git status
On branch master
Your branch is behind 'origin/master' by 4 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)

nothing to commit, working tree clean
```



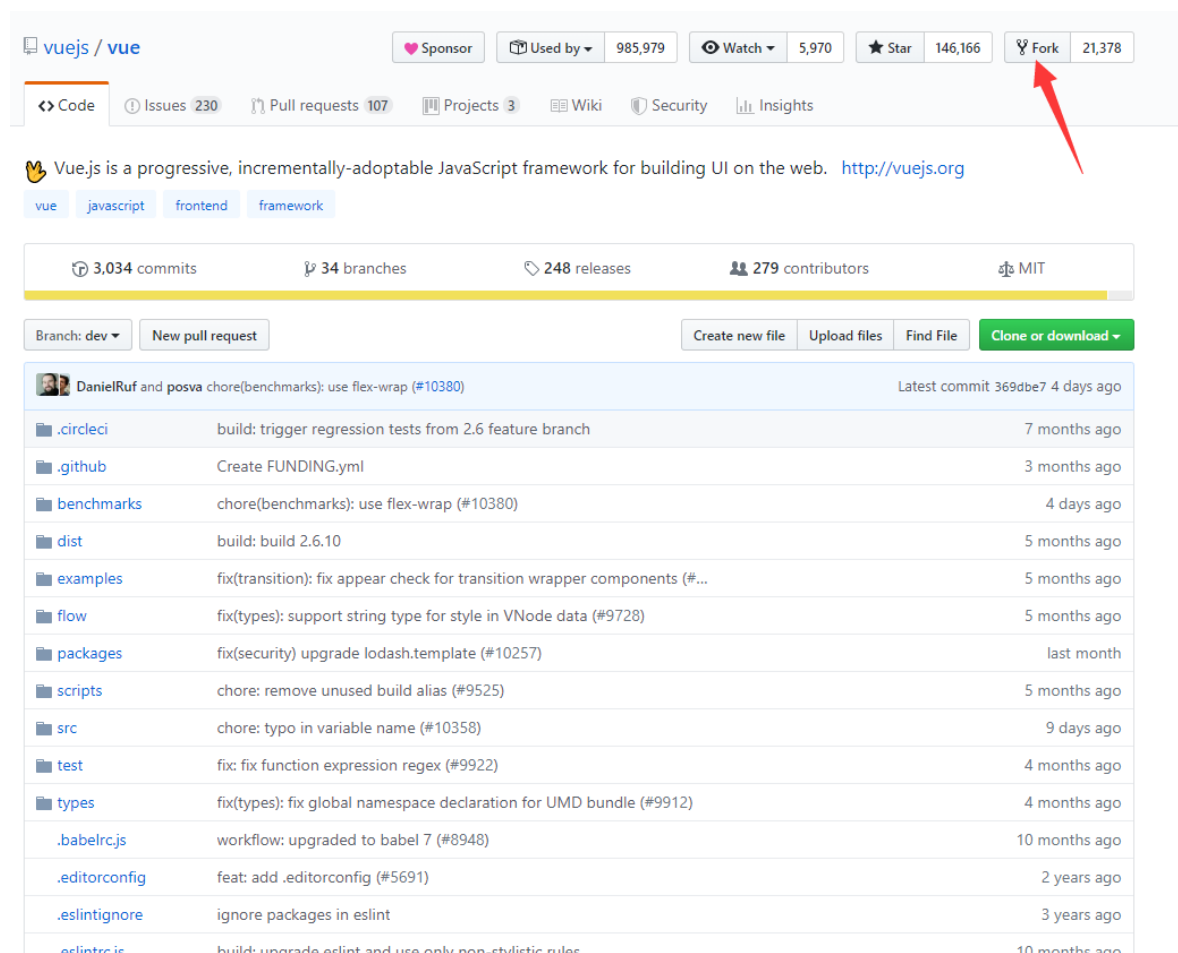
github上也会显示出版本号出来

## 小结

- 命令 `git tag <tagname>` 用于新建一个标签，默认为 HEAD，也可以指定一个commit id；
- 命令 `git tag -a <tagname> -m "blablabla..."` 版本号 可以指定标签信息；
- 命令 `git tag` 可以查看所有标签。
- 命令 `git push origin <tagname>` 可以推送一个本地标签；
- 命令 `git push origin --tags` 可以推送全部未推送过的本地标签；
- 命令 `git tag -d <tagname>` 可以删除一个本地标签；
- 命令 `git push origin :refs/tags/<tagname>` 可以删除一个远程标签。

## 关于自己的远程仓库

我们可以在github上看到很多其他大牛的代码库, 我们通过点击Fork( 叉, 分叉 )来把这个库放在自己的库里面, 这样就相当于我们自己的库里面有了一个完整的项目代码, 然后Git clone到本地, 我们可以对自己的库里所保存的备份代码进行维护和升级, 但是没法对原始项目进行维护



The screenshot shows the GitHub repository page for `vuejs/vue`. The repository has 985,979 uses, 5,970 watches, 146,166 stars, and 21,378 forks. A red arrow points to the 'Fork' button. Below the repository header, there are tabs for Code, Issues (230), Pull requests (107), Projects (3), Wiki, Security, and Insights. The description states: 'Vue.js is a progressive, incrementally-adoptable JavaScript framework for building UI on the web. <http://vuejs.org>'. Below this are tags for 'vue', 'javascript', 'frontend', and 'framework'. The repository statistics show 3,034 commits, 34 branches, 248 releases, 279 contributors, and MIT license. At the bottom, there is a table of recent commits.

Commit	Author	Message	Time
Latest commit 369dbe7	DanielRuf and posva	chore(benchmarks): use flex-wrap (#10380)	4 days ago
7 months ago		build: trigger regression tests from 2.6 feature branch	
3 months ago		Create FUNDING.yml	
4 days ago		chore(benchmarks): use flex-wrap (#10380)	
5 months ago		build: build 2.6.10	
5 months ago		fix(transition): fix appear check for transition wrapper components (#...	
5 months ago		fix(types): support string type for style in VNode data (#9728)	
last month		fix(security) upgrade lodash.template (#10257)	
5 months ago		chore: remove unused build alias (#9525)	
9 days ago		chore: typo in variable name (#10358)	
4 months ago		fix: fix function expression regex (#9922)	
4 months ago		fix(types): fix global namespace declaration for UMD bundle (#9912)	
10 months ago		workflow: upgraded to babel 7 (#8948)	
2 years ago		feat: add .editorconfig (#5691)	
3 years ago		ignore packages in eslint	
10 months ago		build: upgrade eslint and use only non-stylistic rules	

我们自己就有了一份, 然后在克隆到本地

naturewind19 / vue  
forked from vuejs/vue

Watch 0 Star 0 Fork 21,379

Code Pull requests 0 Projects 0 Wiki Security Insights Settings

Vue.js is a progressive, incrementally-adoptable JavaScript framework for building UI on the web. <http://vuejs.org>

Manage topics

3,034 commits 34 branches 248 releases 279 contributors MIT

Branch: dev New pull request

Create new file Upload files Find File Clone or download

This branch is even with vuejs:dev.

DanielRuf and posva chore(benchmarks): use flex-wrap (vuejs#10380)

.circleci build: trigger regression tests from 2.6 feature branch

.github Create FUNDING.yml

benchmarks chore(benchmarks): use flex-wrap (vuejs#10380)

dist build: build 2.6.10

examples fix(transition): fix appear check for transition wrapper components (v...

flow fix(types): support string type for style in VNode data (vuejs#9728)

packages fix(security) upgrade lodash.template (vuejs#10257)

scripts chore: remove unused build alias (vuejs#9525)

src chore: typo in variable name (vuejs#10358)

test fix: fix function expression regex (vuejs#9922)

types fix(types): fix global namespace declaration for UMD bundle (vuejs#9912)

.babelrc.js workflow: upgraded to babel 7 (vuejs#8948)

Clone with SSH ? Use HTTPS

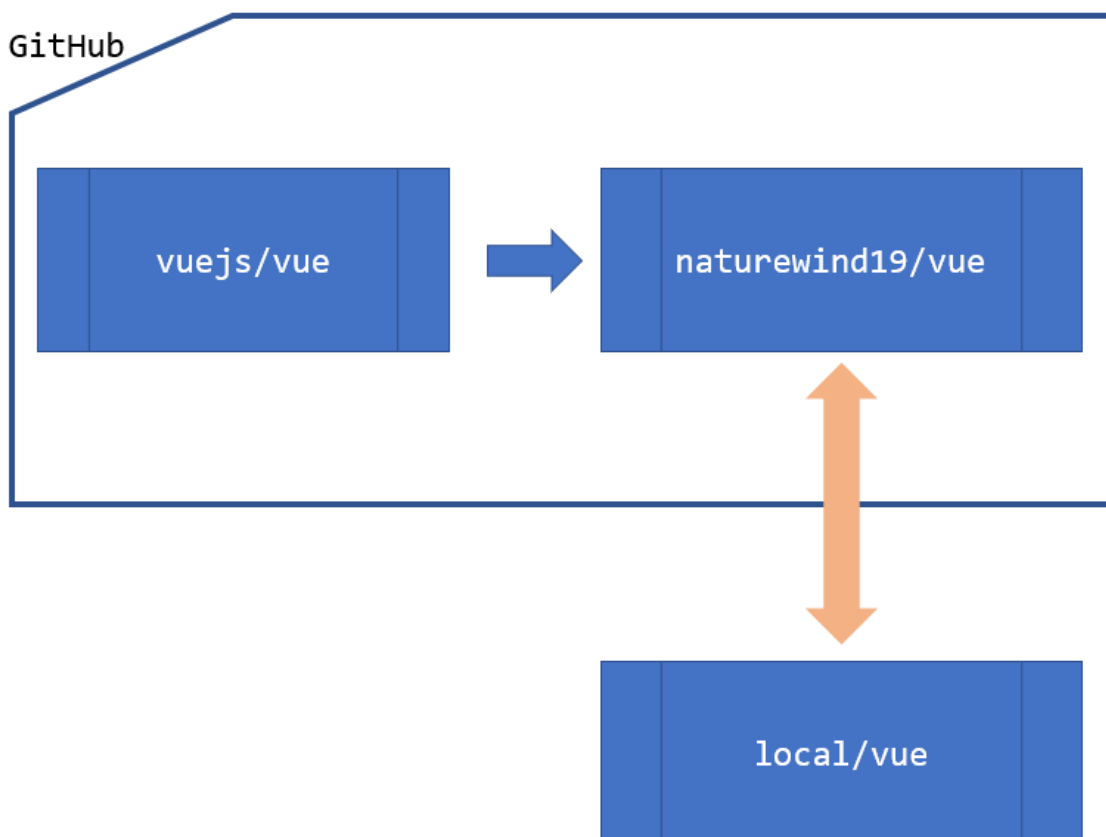
Use an SSH key and passphrase from account.

git@github.com:naturewind19/vue.git

Open in Desktop Download ZIP

```
λ git clone git@github.com:naturewind19/vue.git
Cloning into 'vue'...
remote: Enumerating objects: 55982, done.
Receiving objects: 13% (7777/55982), 2.82 MiB | 16.00 KiB/s
```

### 几个库之间的关系图



如果你想修复vue的一个bug，或者新增一个功能，立刻就可以开始干活，干完后，往自己的仓库推送。

如果你希望vue的官方库能接受你的修改，你就可以在GitHub上发起一个pull request。当然，对方是否接受你的pull request就不一定了。

当你想更正别人仓库里的错误时，要走一个流程：

1. 先 fork 别人的仓库，相当于拷贝一份，相信我，不会有人直接让你改修原仓库的
2. clone 到本地分支，做一些 bug fix
3. 提交到自己的远程fork来的库
4. 发起 pull request 给原仓库，让他看到你修改的 bug
5. 原仓库 review 这个 bug，如果是正确的话，就会 merge 到他自己的项目中