

git教程(5) 标签管理

一 创建标签【创建标签的作用??!!!】

1 在 git 中打标签很简单。

1.1 线切换到要打标签的分支。git checkout master。

```
$ git branch
* dev
  master

$ git checkout master
Switched to branch 'master'
```

1.2 输入 git tag tag_name。即可为“当前活跃分支”打上标签。

```
$ git tag v1.0
```

git tag 【为啥不是 git tag list 与 git stash list相对应，当然说是与 git log 相对应也是可以的!!】

注意：默认 标签是打在 最新提交的 commit上的。

当然我们可以找到相应提交的 id，然后给他打上标签【git tag 要补充的标签名字 commit_id】：

比方说要对 `add merge` 这次提交打标签，它对应的commit id是 `f52c633`，敲入命令：

```
$ git tag v0.9 f52c633
```

1.3 git tag【标签是按字母排序的，不是按时间!!】

```
$ git tag
v0.9
v1.0
```

注意：查看相关标签的详细信息【git show tag_name】

```
$ git show v0.9
commit f52c63349bc3c1593499807e5c8e972b82c8f286 (tag: v0.9)
Author: Michael Liao <askxuefeng@gmail.com>
Date:   Fri May 18 21:56:54 2018 +0800

    add merge

diff --git a/readme.txt b/readme.txt
...
```

1.4 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！！【如 git tag -a '标签名' -m '相应的标签说明信息' commit_id】

```
$ git tag -a v0.1 -m "version 0.1 released" 1094adb
```

用命令 `git show <tagname>` 可以看到说明文字：

```
$ git show v0.1
tag v0.1
Tagger: Michael Liao <askxuefeng@gmail.com>
Date:   Fri May 18 22:48:43 2018 +0800

version 0.1 released

commit 1094adb7b9b3807259d8cb349e7df1d4d6477073 (tag: v0.1)
Author: Michael Liao <askxuefeng@gmail.com>
Date:   Fri May 18 21:06:15 2018 +0800

    append GPL

diff --git a/readme.txt b/readme.txt
...
```

！！！注意：标签总是会和某个commit挂钩的。如果这个 commit 既出现在 master ,又出现在 dev分支上，那么这2分支都会看到这个标签！！

2 小结

2.1 命令 git tag tag_name 用于创建一个标签【默认该标签打在HEAD上】，当然我们可以

通过commit_id 将该标签打到 相应的提交上【这样一来，共用同一个提交的多个分支都会被打上这个标签。】！！

2.2 git tag -a tag_name -m '标签信息'。-a参数指定 要打上的标签名， -m指定 标签的信息！！

2.3 git tag查看所有的标签【不是 git tag list！！！ 没有 list！！！！】

2.4 查看某标签的 详细信息 git show tag_name

二 操作标签

1 删除标签【git tag -d tag_name】

```
$ git tag -d v0.1
Deleted tag 'v0.1' (was f15b0dd)
```

注意：因为创建的标签只会存储在本地，不会自动推送到远程，我们可以安全地将标签删除！！

如果要推送某个标签 到远程 ---> git push origin tag_name 。

```
$ git push origin v1.0
Total 0 (delta 0), reused 0 (delta 0)
To github.com:michaelliao/learngit.git
* [new tag]          v1.0 -> v1.0
```

一次性推送 将 尚未推送到远程 的本地标签 ---> git push origin --tags【其实就是将 具体的 tag_name 换成了 --tags！！！！】

```
$ git push origin --tags
Total 0 (delta 0), reused 0 (delta 0)
To github.com:michaelliao/learngit.git
* [new tag]          v0.9 -> v0.9
```

1.2 如果已经推送到远程的标签，现在要进行删除。较麻烦，应从本地中先删除！！！！【git tag -d tag_name】

```
$ git tag -d v0.9
Deleted tag 'v0.9' (was f52c633)
```

接着再从远程中删除。删除命令也是 push .格式较复杂【git push origin :refs/tags/tag_name !!!】

```
$ git push origin :refs/tags/v0.9
To github.com:michaelliao/learngit.git
- [deleted]          v0.9
```

2 小结

2.1 推送一个本地标签 git tag origin tag_name

2.2 推送所有 未推送到远程的标签 git push origin --tags

2.3 删除本地标签（未推送到远程分支） git tag -d tag_name

2.4 删除远程上的标签 git push origin :refs/tags/tag_name