在Git中打标签非常简单，首先，切换到需要打标签的分支上：

$ git branch

\* dev

master

$ git checkout master

Switched to branch 'master'

然后，敲命令git tag <name>就可以打一个新标签：

$ git tag v1.0

可以用命令git tag查看所有标签：

$ git tag

v1.0

默认标签是打在最新提交的commit上的。有时候，如果忘了打标签，比如，现在已经是周五了，但应该在周一打的标签没有打，怎么办？

方法是找到历史提交的commit id，然后打上就可以了：

$ git log --pretty=oneline --abbrev-commit

12a631b (HEAD -> master, tag: v1.0, origin/master) merged bug fix 101

4c805e2 fix bug 101

e1e9c68 merge with no-ff

f52c633 add merge

cf810e4 conflict fixed

5dc6824 & simple

14096d0 AND simple

b17d20e branch test

d46f35e remove test.txt

b84166e add test.txt

519219b git tracks changes

e43a48b understand how stage works

1094adb append GPL

e475afc add distributed

eaadf4e wrote a readme file

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

再用命令git tag查看标签：

$ 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。可以用git show <tagname>查看标签信息：

$ 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

...

可以看到，v0.9确实打在add merge这次提交上。

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：

$ 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。

### git tag <tagname>用于新建一个标签，默认为HEAD，也可以指定一个commit id；

### git tag -a <tagname> -m "blablabla..."可以指定标签信息；

### git tag可以查看所有标签。

### git show <tagname>查看标签信息：