首先,master 分支应该是非常稳定的,也就是仅用来发布新版本,平时不能在上面干活;

那在哪干活呢?干活都在 dev 分支上,也就是说,dev 分支是不稳定的,到某个时候,比如 1.0 版本发布时,再把 dev 分支合并到 master上,在 master 分支发布 1.0 版本;

你和你的小伙伴们每个人都在 dev 分支上干活,每个人都有自己的分支,时不时地往 dev 分支上合并就可以了。

开发一个新功能,最好新建一个分支;

如果要丢弃一个没有被合并过的分支,可以通过 git branch -D <name>强行删除。

多人协作的工作模式通常是这样:

- 1. 首先,可以尝试用 git push origin <branch-name>推送自己的修改;
- 2. 如果推送失败,则因为远程分支比你的本地更新,需要先用 git pull 试图合并;
- 3. 如果合并有冲突,则解决冲突,并在本地提交;
- 4. 没有冲突或者解决掉冲突后,再用 git push origin

 / branch-name >推

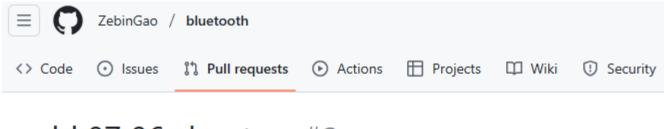
 送就能成功!

检查环境是否同步

admin@zebin MINGW64 /d/BlueTestTry/bluetooth (master)

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.
nothing to commit, working tree clean
检出dev分支,本地的名字可以和远程分支的名字不一样
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (master)
$ git checkout -b gao origin/dev
Switched to a new branch 'gao'
Branch 'gao' set up to track remote branch 'dev' from 'origin'.
进行修改,添加并提交
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (gao)
$ git add .
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (gao)
$ git commit -m "add 07 06 chapters"
[gao 1104fdf] add 07 06 chapters
推送到远程的分支
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (gao)
$ git push origin gao:dev
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
确认一遍
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (gao)
$ git push origin gao:dev
Everything up-to-date
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (gao)
$ git switch master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
切换到本地master并删除分支
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (master)
$ git branch -d gao
```

下面是管理员的操作, merge 后的样子。



add 07 06 chapters #2

