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

warning: deleting branch 'gao' that has been merged to

'refs/remotes/origin/dev', but not yet merged to HEAD.

Deleted branch gao (was 1104fdf).

下面是管理员的操作，merge后的样子。

