



链滴

Git之分支创建策略

作者: [justdoit](#)

原文链接: <https://ld246.com/article/1469769454096>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>分支策略：git上始终保持两个分支，master分支与develop分支。master分支主要用于发布时使用，而develop分支主要用于开发使用。

创建master的分支develop
git checkout -b develop master

换到master分支
git checkout master

合并develop分支到master
git merge --no-ff develop

除了以上两个常驻分支外，我们还可以适当分支出三种分支：功能分支、预发布分支、修补分支，这三种分支使用完后也该删除，保持两个常驻分支。

功能分：该分支从develop中分支出来，开发完成后再合并入develop，名字采用feature*的形式命名。
创建功能分支：
 git checkout -feature-x develop
开发完成后，合并到develop分支：
<s an> git checkout develop
 git merge --no-ff feature-x
最后删除分支:
 git branch -d feature-x
<r />
预发布分支：正是版本发布前，既合并到master分支前，因此预发布分支是从develop分支出来的，预发布后，必修合并进develop和master。命名采用release-*的形式。<r />创建一个预发布分支：
 git checkout -b release-* develo
确认版本没有问题后，合并到master分支：
 git checkout master
 git merge --no-ff release-*
对合并生成的新节点，做一个标签：
 git tag -a 1.2
再合并到develop分支:
 git checkout decelop</pan>
 git merge --no-ff release-*
最后删除分支:</spa>
 git branch -d release-*

修补分：主要用于修改bug的分支，从master分支分出来，修补后，在合并进master和develop分支。命名用fixbug-*形式。
创建一个修补分支：
 git che kout -b fixbug-* master
修补结束后,合并到master分支:
 git checkout master
 git merge --no-ff fixbug-*
 git tag -a 0.1.1
再合并到develop分支:
 git checkout develop
 git merge --no-f fixbug-*最后删除分支:
 git branch -d fixbug-*</p>