

为何迁移到 Coding ?

Coding 的页面托管服务

提交多个分支

Atom/git/coding 的三角关系

第五章：分支专题

 简单分支操作

简单分支操作

今天的主角是分支，因为不介绍分支（branch）的概念，下面的操作是没办法介绍了。可以这样说，Git 最核心的操作对象是版本（commit），最核心的操作技巧就是分支。

什么是分支？

仓库创建后，一旦有了新 `commit`，默认就会放到一个分支上，名字叫 `master`。前面咱们一直看到的多个版本组成的一条历史线，就是 `master` 分支。但是一个仓库内，用户可以自己创建其他的分支，可以有多条历史线。

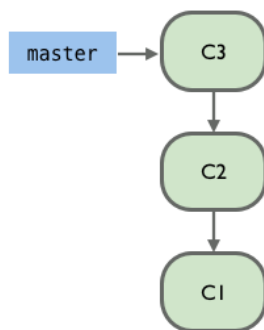
说说 `master` 这个名字，一般中文叫“主分支”，其实从技术底层来讲它跟其他我们自己要创建的分支没有区别，只不过它是天生的默认分支。实际工程项目中会人为的给它一个重要的使命，存放稳定代码。就像 `github` 公司倡导的。

`master` 分支上的所有代码都应该是可以部署的

意思就是 `master` 分支上的代码是随时可以放到产品服务器上跑的代码。这样，如果想开发一个新功能，可以新开分支。想象一下历史线上有很多节，每个版本就是一节。一个分支相当于一跟竹子，一节节的往上长。



但是实际上在底层并不是每个分支都拷贝出自己独立的一条历史线。其实 `master` 本身只是一个指针，指向 `master` 分支上最新的一个版本。这样由于每个 `commit` 都可以顺藤摸瓜找到自己的前一个 `commit`，那么这条历史线就可以确定了。



创建新分支

什么时候需要开一个新分支，这个后面讲各种工作流程的时候会介绍，今天先把基本操作学会。

执行命令

```
git checkout -b idea
```

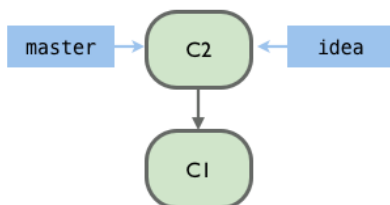
就可以创建一个名字叫 `idea` 的分支了。默认创建的这个分支跟 `master` 分支内容相同。

执行

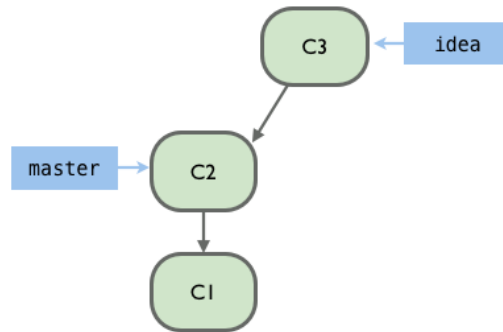
```
git branch
```

可以看到“当前分支”（`current branch`）已经自动切换到 `idea` 分支。

但是，在底层这个的实现是非常巧妙的，就是又创建一个新的 `idea` 指针，跟 `master` 指针指向同一个版本，根本没有拷贝历史线。



如果现在我对项目做一下修改，然后 commit 了。那么移动的只是 idea 指针，master 不变。就成了这样：

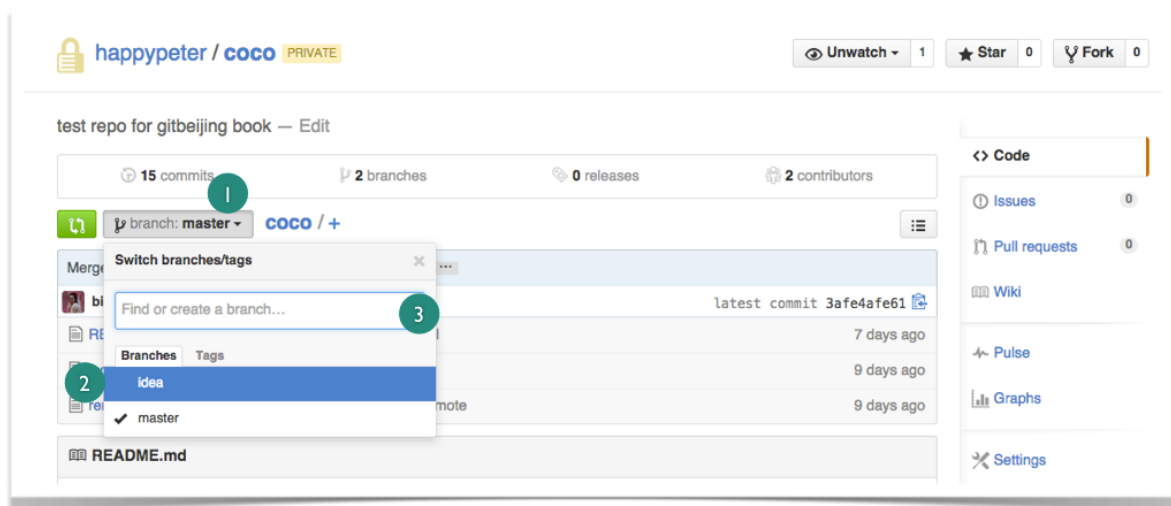


现在 master 分支包含两个版本 C1 和 C2，idea 分支包含三个版本 C1，C2，C3。

默认情况下这个 idea 分支只是存在于本地，如果想在远端仓库上发布这个分支，执行

```
git push origin idea
```

这样，到远端仓库看一下，点击下图1处，发现果然2处多了一个 idea 分支，3处的输入框中，不但能搜索已有分支，还能创建新分支，看到了吧，很多操作在本地客户端和 github.com 上都能进行。



切换分支

当前分支如果是 idea，那想要切换回 master，只需要执行

```
git checkout master
```

注意，每次切换分支，项目代码，术语叫工作树（Working Tree）是会随着变化的，在编辑器中看看就知道了。

删除分支

在客户端把分支切换到 master，现在试图去删除 idea，执行

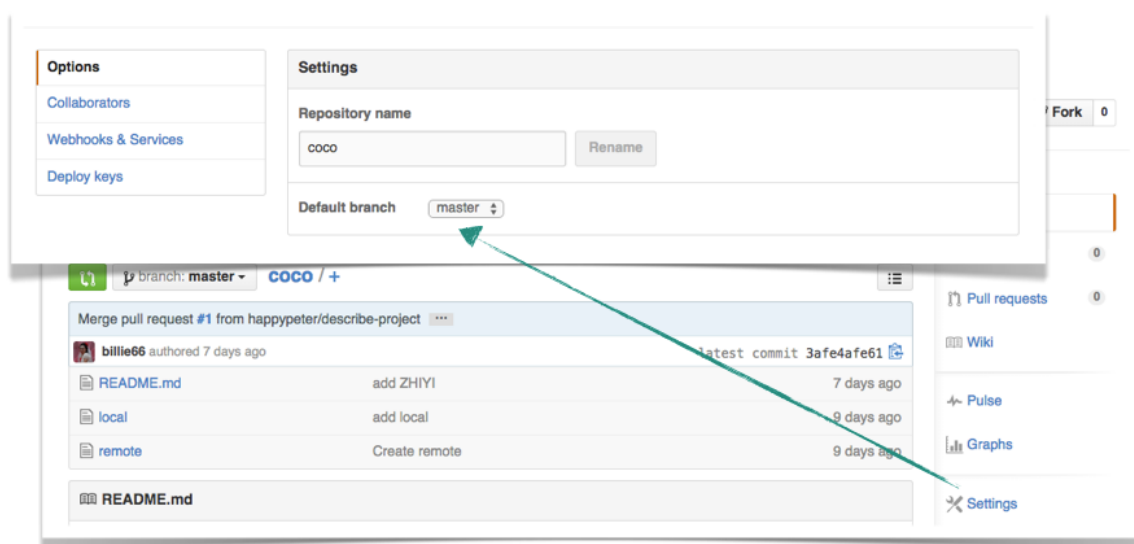
```
git branch -D idea
```

如果想要删除 github 上的 idea 分支

```
git push origin :idea
```

就可以了。

注意，github 上已经被设置为“默认分支”的分支是不能被删除的，master 一般就是默认分支。在远端仓库，也就是 github.com 上如何切换默认分支呢？到 settings 下面就更改 Default branch 就可以了。



总结

只开测试分支，调好代码 `commit` 了之后，如果不把代码搞到 `master` 分支上是没有太大意义的，这就涉及到分支合并的问题了，这个是 `git` 最大最强的一块功能，后面介绍。

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3