杰普软件科技有限公司
www.briup.com

Tel: (021)55660810
Fax: (021)55660802
Email:training@briup.com
Msn: training.sh@hotmail.com
Home:http://www.briup.com

地址：上海市闸北区万荣路 1188 **弄 F栋 6**号三层 **-** 上海服务外包科技园龙软园区
邮编： 200436

电话： 021-56657112
传真： 021-55661523-8003
电邮： training@briup.com
主页：
http://www.briup.com

Briup High-End IT Training

Node

第 1 章：git

Brighten Your Way And Raise You Up.

- 介绍

Git 是目前世界上最先进的分布式版本控制系统

下载与安装:

$ sudo apt-get install git

◆ 创建本地仓库（**repository**）

repository，你可以简单理解成一个目录，这个目录里面的所有文件都可以被 Git 管理起来，每个文件的修改、删除， Git 都能跟踪，以便任何时刻都可以追踪历史，或者在将来某个时刻可以“还原”，注意，所有的版本控制系统，其实只能跟踪文本文件的改动，比如 TXT 文件，网页，所有的程序代码等。

$ mkdir app

$ cd app

$ git init

Initialized empty Git repository in /home/briup/app/.git/

◆ 管理文件

第一步，用命令 git add 告诉 Git，把文件添加到仓库。

$ vim readme.txt // 创建文件并且编辑它

$ git add readme.txt

第二步，用命令 git commit 告诉 Git 把文件提交到仓库，同时使用 -m 指定信息

$ git commit -m "wrote a readme file”

使用 git status 命令可以让我们时刻掌握仓库当前的状态

$ git status

使用 git diff 命令查看对仓库进行了哪些具体的修改

$ git diff

- 版本管理

使用 git log 命令显示从最近到最远的提交日志。

$ git log

```
commit fe1129a9f5f7374de84f1ad5dbc9ea87a19db2ca
Author: briup licy@briup.com
Date:   Wed Sep 20 06:21:54 2017 +0800
day02

commit be91b171b6eadb834dd2a9773d944dcbececfa78
Author: briup licy@briup.com
Date:   Wed Sep 20 06:18:56 2017 +0800
day01
```

使用 git reset 命令可以将版本回退到任意节点。

$ git reset –hard be91b1

使用 git reflog 命令可以查看历史命令。

$ git reflog

- 文件删除

$ rm readme.txt // 删除文件

$ git status // 查看状态

这时你有两个选择，一是确实要从版本库中删除该文件，那就用命令 git rm 删掉，并且 git commit：

$ git rm readme.txt

$ git commit –m 'confirm delete'

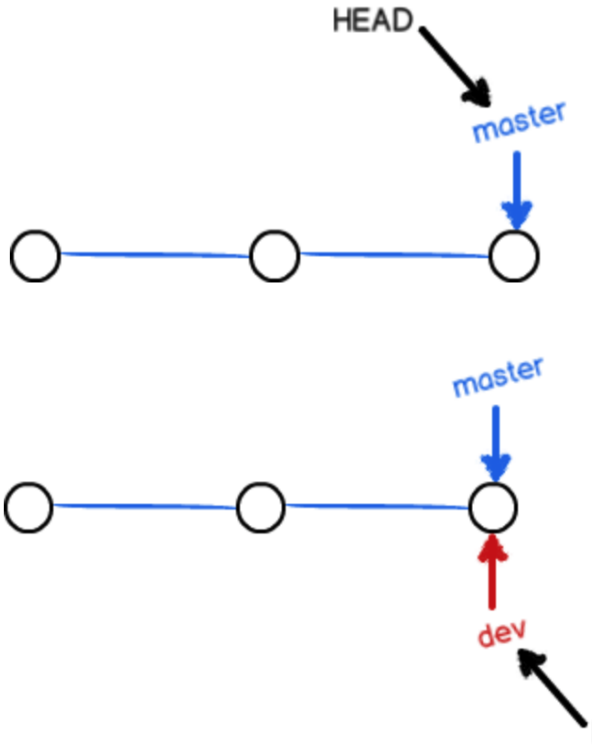
使另一种情况是删错了，因为版本库里还有呢，所以可以很轻松地把误删的文件恢复到最新版本。

$ git checkout -- readme.txt

◆ 分支管理

在 Git 中，每次提交， Git 都把它们串成一条时间线，这条时间线就是一个分支。截止到目前，只有一条时间线，在 Git 里，这个分支叫主分支，即 master 分支。 HEAD 严格来说不是指向提交，而是指向 master ， master 才是指向提交的，所以， HEAD 指向的就是当前分支。

当我们创建新的分支 dev 时， Git 新建了一个指针叫 dev ，指向 master 相同的提交，再把 HEAD 指向 dev ，就表示当前分支在 dev 上：



$ git branch dev *// 创建新分支* **dev**

$ git checkout dev *// 切换至* **dev**

$ git branch *// 查看当前分支状态*

* dev

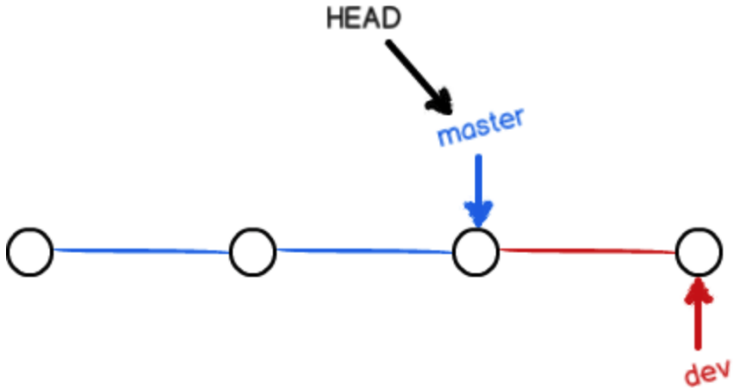
master

- 分支管理

如果此时切换到主分支上，将看不到 dev 分支上的修改

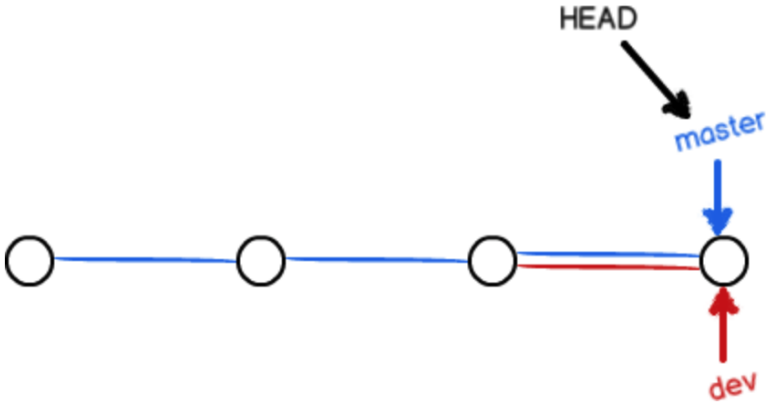
$ git checkout master

Switched to branch 'master'



此时我们把 dev 分支的工作成果合并到 master 分支上

$ git merge dev



远程仓库

可以在 github.com 中注册一个账号，就可以创建远程仓库了。创建成功后会产生一个远程仓库地址， 例如： https://github.com/pluslicy/app.git ， 我们可以将本地代码提交到远程仓库中。远程库的名字就是 origin ， 这是 Git 默认的叫法。

$ git remote add origin https://github.com/pluslicy/app.git

下一步，就可以把本地库的所有内容推送到远程库上：

$ git push origin master

也可以将其他分支推送到远程仓库上：

$ git push origin dev

- 注意：如果远程仓库中有初始化文件，在提交之前要先更新

$ git pull origin master

- 远程仓库

可以使用 git clone 命令来克隆其他小伙伴仓库中的内容

$ git clone https://github.com/pluslicy/app.git

从远程库 clone 时，默认情况下，你的小伙伴只能看到本地的 master 分支，你的小伙伴要在 dev 分支上开发，就必须创建远程 origin 的 dev 分支到本地

$ git checkout -b dev origin/dev