

学习 Git 总结

一、笔记时空

时间：2017 年 11 月 27 日

地点：浙江工业大学致远楼阳光长廊

用时：一个下午

二、学习感悟

今天很高兴 get 到一个新技能，即学习 git 分布式版本控制系统（distributed version control system）。这个软件最大的特点是免费、轻量、快捷，大概花了一个下午的时间搞清楚基础操作。

三、学习目录

Git 的主要功能有本地仓库管理、远程仓库管理和团队协作三个部分。

四、仓库配置（Repository）

首先，安装完 Git 后，要配置帐号，可以使用：

```
$ git config --global user.name "Your Name"
```

```
$ git config --global user.email "email@example.com"
```

然后，我们需要创建一个版本库，版本库又名仓库，英文名 **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”。代码如下：

```
$ mkdir learngit
$ cd learngit
$ pwd
$ git init
```

五、本地仓库管理

5.1 文件上传到版本库

本地仓库分工作区，暂存区和仓库区，如下图所示：

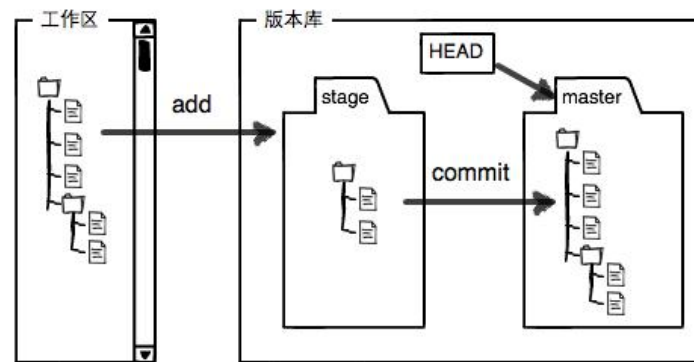


图 1 版本库工作区间

上传文件到版本库中需要两步走，如下：

当添加文件到目录时，相当于把文件放进工作区；

第一步，`$git add filename`：把文件加入暂存区；

第二步，`$git commit -m 'description'`：把暂存区放入版本库中。

5.2 修改文件版本

`$git reset --hard^`:返回上个版本，`hard^^`上两个，`hard~num`上 num 个。

`$git reset --hard commit_id`:返回提交 ID 号

`$git log` 查询提交历史，确定返回版本

`$git reflog` 查看命令历史

`$git checkout -- name`:用版本库替代工作区，用于误删或工作区误修改

```
$git rm name:删除
```

5.3 远程服务器

本地关联远程服务器:

在 Github 上创建一个仓库，需要用本地的 SSH key，注意的是，Github 虽然是免费托管服务器，但是任何人都可以看到，不要把敏感信息放进去。

要关联一个远程库，使用命令 `$git remote add origin git@server-name:path/repo-name.git`。

关联后，使用命令 `$git push -u origin master` 第一次推送 master 分支的所有内容。此后，每次本地提交后，只要有必要，就可以使用命令 `$git push origin master` 推送最新修改。

分布式版本系统的最大好处之一是在本地工作完全不需要考虑远程库的存在，也就是有没有联网都可以正常工作，而 SVN 在没有联网的时候是拒绝干活的！当有网络的时候，再把本地提交推送一下就完成了同步，真是太方便了！

远程服务器 Clone:

```
$ git clone git@github.com:michaelliao/gitskills.git
```

要克隆一个仓库，首先必须知道仓库的地址，然后使用 `git clone` 命令克隆。Git 支持多种协议，包括 https，但通过 ssh 支持的原生 git 协议速度最快。

5.4 分支管理

分支管理的关系图如下所示：

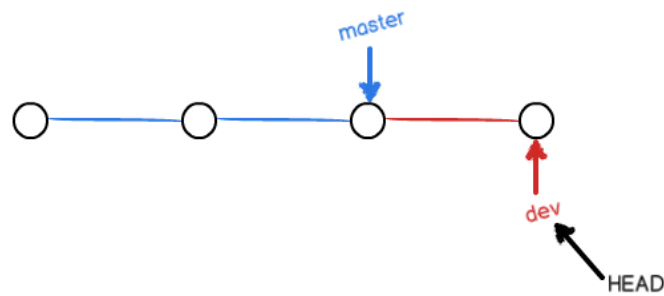


图 2 分支管理关系图

由于 Git 采用指针的方式管理分支，因此创建、合并和删除分支特别快！

Git 鼓励大量使用分支：

查看分支：`git branch`

创建分支：`git branch <name>`

切换分支：`git checkout <name>`

创建 + 切换分支：`git checkout -b <name>`

合并某分支到当前分支：`git merge <name>`

删除分支：`git branch -d <name>`

无法自动合并分支时，就必须首先解决冲突。解决冲突后，再提交，合并完成。用`$ git log --graph --pretty=oneline --abbrev-commit`命令可以看到分支合并图。

当手头工作没有完成时，先把工作现场 `git stash` 一下，然后去修复 bug，修复后，再 `git stash pop`，回到工作现场。

推送 Branch 到远程服务器：

查看远程库信息，使用 `git remote -v`；

本地新建的分支如果不推送到远程，对其他人就是不可见的；

从本地推送分支，使用`$git push origin branch-name`，如果推送失败，先用 `git pull` 抓取远程的新提交；

在本地创建和远程分支对应的分支，使用`$git checkout -b branch-name origin/branch-name`，本地和远程分支的名称最好一致；

建立本地分支和远程分支的关联，使用`$git branch --set-upstream branch-name`

`origin/branch-name`;

从远程抓取分支，使用 `git pull`，如果有冲突，要先处理冲突。

5.5 标签管理 (Tag)

命令 `$git tag <name>` 用于新建一个标签，默认为 HEAD，也可以指定一个 commit id;

`$git tag -a <tagname> -m "blablabla..."` 可以指定标签信息;

`$git tag -s <tagname> -m "blablabla..."` 可以用 PGP 签名标签;

命令 `$git tag` 可以查看所有标签。

5.6 使用 GitHub

在 GitHub 上，可以任意 Fork 开源仓库;

自己拥有 Fork 后的仓库的读写权限;

可以推送 pull request 给官方仓库来贡献代码。

六、自定义 Git

`$ git config --global color.ui true`

使命令的显示更加醒目!