**Git学习总结**

—— 李宣廷

**1.git介绍**

Git是分布式版本控制系统，那么它就没有中央服务器的，每个人的电脑就是一个完整的版本库，这样，工作的时候就不需要联网了，因为版本都是在自己的电脑上。既然每个人的电脑都有一个完整的版本库。比如说自己在电脑上改了文件A，其他人也在电脑上改了文件A，这时，你们两之间只需把各自的修改推送给对方，就可以互相看到对方的修改了。

**2.git配置**

git config --global user.name “lixuanting”

git config --global user.email “xuanting.li@neoway.com”

**3.创建版本库**

在桌面新建了一个testgit文件夹，右击打开git bash here，通过git init把这个目录变成git可以管理的仓库。这是testgit文件夹下多了一个隐藏的.git文件夹。

**4.工作区和暂存区**

工作区：就是在电脑上看到的目录，比如目录下testgit里的文件(.git隐藏目录版本库除外)。或者以后需要再新建的目录文件等等都属于工作区范畴。  
版本库(Repository)：工作区有一个隐藏目录.git,这个不属于工作区，这是版本库。其中版本库里面存了很多东西，其中最重要的就是stage(暂存区)，还有Git为我们自动创建了第一个分支master,以及指向master的一个指针HEAD。

Git提交文件到版本库有两步：

第一步：是使用 git add 把文件添加进去，实际上就是把文件添加到暂存区。

第二步：使用git commit提交更改，实际上就是把暂存区的所有内容提交到当前分支上。

**5.练习git**

在testgit文件夹下创建了readme.txt，内容为11111111111

将readme.txt添加到暂存区：git add readme.txt

提交readme.txt：git commit –m “提交readme.txt”

通过git status查看是否还有文件未提交，显示nothing to commit, working directory clean，表示没有文件未提交。

修改readme.txt，加一行222222222222，使用git status，有红色字体提示：

modified: readme.txt

使用git diff readme.txt，查看修改了什么。

使用git log可以查看修改日志。

使用git reset --hard HEAD^,可以退回上个版本。使用git reset --hard HEAD^^,可以退回上上个版本。

使用cat readme.txt可以看到，文件已退回修改前版本。

版本如果过多，想退回的指定的一个版本。先使用git reflog，查看历史版本号。

然后用git reset --hard 版本号，退回指定的版本。

使用git checkout --file可以放弃工作区的修改

删除文件使用：rm readme.txt

想在版本库删除此文件，使用commit命令

想恢复此文件，使用git checkout --readme.txt

创建分支dev：git branch dev

创建并进入分支：git checkout -b dev 相当于git branch dev和git checkout dev

git branch 查看分支，列出所有的分支，当前的分支前面会添加一个星号。在dev分支修改了文件，master分支上的文件不会改变，切回master后：git checkout master，使用git merge dev在当前分支上合并dev分支的内容。

删除分支的命令为：git branch –d name  
 **远程仓库：**

先注册github账号，然后在本地户主目录下创建SSH Key，登陆github，Add SSH Key。

在github上创建一个仓库，然后在本地户主目录下运行：git remote add origin +远程仓库地址.git

在本地用git push origin master，将当前分支master推送到远程。

从远程仓库克隆命令：git clone +远程仓库地址

**6.培训git学习内容**

Git diff

　　 　　用于比较两次修改的差异

　　　　1.1 比较工作区与暂存区

　　　　　　git diff 不加参数即默认比较工作区与暂存区

　　　　1.2 比较暂存区与最新本地版本库（本地库中最近一次commit的内容）

　　　　　　git diff --cached  [<path>...]

　　　　1.3 比较工作区与最新本地版本库

　　　　　　git diff HEAD [<path>...]  如果HEAD指向的是master分支，那么HEAD还可以换成master

　　　　1.4 比较工作区与指定commit-id的差异

　　　　　　git diff commit-id  [<path>...]

　　　　1.5 比较暂存区与指定commit-id的差异

　　　　　　git diff --cached [<commit-id>] [<path>...]

　　　　1.6 比较两个commit-id之间的差异

　　　　　　git diff [<commit-id>] [<commit-id>]

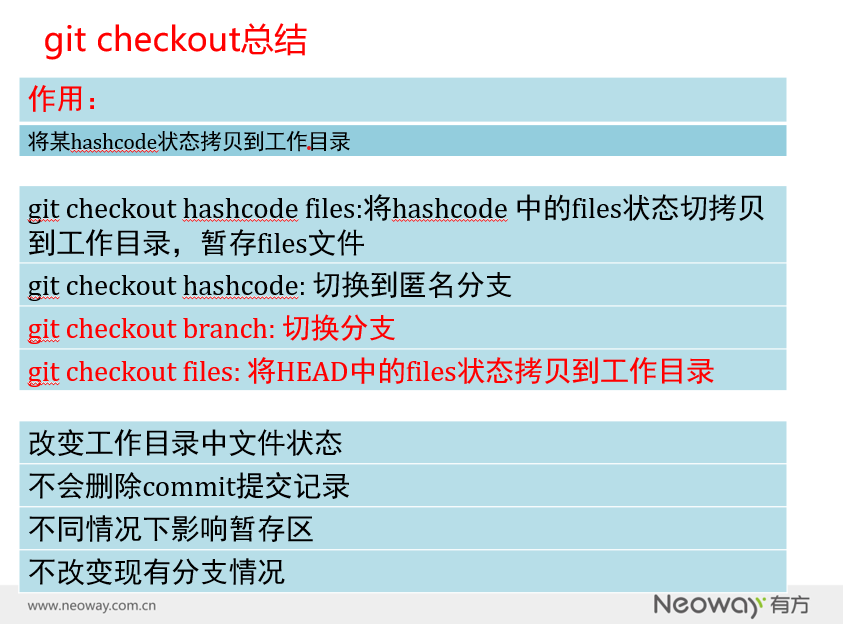
　　　　1.7 使用git diff打补丁

Git commit

        git commit 主要是将暂存区里的改动给提交到本地的版本库。每次使用git commit 命令我们都会在本地版本库生成一个40位的哈希值，这个哈希值也叫commit-id，commit-id在版本回退的时候是非常有用的，它相当于一个快照,可以在未来的任何时候通过与git reset的组合命令回到这里.

git checkout

git checkout命令用于切换分支或恢复工作树文件。git checkout是git最常用的命令之一，同时也是一个很危险的命令，因为这条命令会重写工作区。



git reset作用

移动当前分支到另外一个commit点

实时更新暂存区和工作目录

可不触动工作目录情况下拷贝文件到暂存区

移动当前分支到某commit，删除此commit之后的所有commits

——hard更新工作目录

——soft更新暂存区

git merge作用

创建一个commit，此commit包含其他commits的变化

在merge前，暂存区必须匹配当前的commit，否则什么都不做

git cherry-pick

copy一个commit；在当前分支创建一个新的commit；与被cherry-pick的commit共享信息。

git push & git pull

本机仓库equal，ahead，behind服务器仓库

若无冲突，git push将本地仓库的commit merge到服务器仓库

git pull将服务器仓库merge到本地仓库中

及时使用git pull保持本地仓库和服务器仓库同步

软件经理

1. 新建A70代码库
2. 若需要，新建分支
3. 对新提交的代码进行检查

开发人员

1. git clone仓库
2. 开发前git pull更新仓库
3. 进行开发
4. 编译确保代码无误并不影响其他功能
5. git commit提交代码
6. git push同步代码到远程仓库