**初涉Git**

Git是一款免费、开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。Git是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理。

Git的特点：分布式相比于集中式的最大区别在于开发者可以提交到本地，每个开发者通过克隆（git clone），在本地机器上拷贝一个完整的Git仓库。

Git的优点：1、适合[分布式开发](http://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E5%BC%80%E5%8F%91)，强调个体。2、公共服务器压力和数据量都不会太大。3、速度快、灵活。4、任意两个开发者之间可以很容易的解决冲突。5、离线工作

缺点：1、学习周期相对而言比较长。2、不符合常规思维。3、代码保密性差，一旦开发者把整个库克隆下来就可以完全公开所有代码和版本信息。

**安装Git**

从官网下载msysgit，按默认路径安装即可。安装完成后在开始菜单里找到Git>Git Bash，会弹出一个命令窗类似的东西，就说明安装成功了；

而后就是配置你的用户名和邮箱地址

git config --global user.name “your name”

git config --global user.email “[email@example.com](mailto:email@example.com)”

**创建版本库**

首先你要先创建一个目录，然后所有的git文件都会在这个目录下管理，无论新建删除都可以跟踪，以便于随时还原

mkdir learngit

cd learngit

pwd //命令是显示当前目录

**添加文件**

git init //把这个目录变成可以管理的仓库

git add readme.txt //将文件添加到仓库

git commit –m “wrote a readme file”//将文件提交到仓库，-m后面是提交说明

ps（conmmit 一次可以提交很多文件，所以可以多次add不同的文件）

git status //查看仓库状态

**回退版本**

git diff readme.txt //查看readme.txt具体修改了哪些

git log //查看历史记录

git log –pretty=oneline //添加参数以达到过滤效果

git reset –hard HEAD^ //回退到之前版本，上一个就^,两个就^^,一百个就HEAD~100

git reset –hard HEAD 2240da //根据版本id回退到指定版本 ID不用写全

ps（HEAD 指针指向被选定的版本）

cat readme.txt //查看文件内容

git reflog //显示历史命令

git remote：查看远端服务器别名，加上-v显示url信息

git remote add server\_url local-alias：添加远端服务器

git clone project\_url local\_alias：拷贝项目到本机中

git push server\_url/local-alias：更新远端服务器仓库

git pull server\_url/local-alias：更新本地服务器仓库

Working Directory(工作区):

Repository(版本库):

Add把文件放进stage（暂缓区），commit（暂缓区文件提交到当前分支）

git diff HEAD – readme.txt命令可以看出暂缓区与版本库里最新版本的区别

get checkout – readme.txt回到最近一次git commit或者gitadd时的状态

git reset HEAD readme.txt 把暂存区修改撤销，重放回工作区

rm test.txt删除文件

git rm test.txt 删除git中保存的文件

**远程仓库**

Ssh-keygen –t rsa –C “myemail@exampe.com” //新建公钥私钥

.ssh目录里，id\_rsa是私钥，id\_rsa.pub公钥

git remote add origin [git@github.com:xxx/xxx //](mailto:git@github.com:xxx/xxx%20//)将本地仓库关联至github

git push –u origin master //本地第一次上传至github

//-u 不止会上传还会把两端的master分支关联

git clone [git@github.com:xxxxxx/xxx.git//](mailto:git@github.com:xxxxxx/xxx.git//)从远程仓库下载

**分支管理**

**创建与合并分支**

git branch dev//创建分支dev

git checkout dev//切换分支dev

git checkput –b dev//创建并切换分支dev

git branch //查看分支

git merge dev //将分支dev合并至主线程

git branch –d //删除分支dev

**解决冲突**

git log//查看日志

git log –graph –pretty=oneline –abbrev-commit//带参数查日志

**分支管理策略**

//master为生产分支，dev为开发主分支，每个各自拥有自己分支。

Git merge –on-ff –m “xxx” dev//--on-ff表示禁用fast forward

标签管理

git tag v1.0//打上标签

git show v1.0//查看标签信息

git tag –d v1.0//删除标签

git push orgin v1.0//推送本地标签

git push origin –tags //推送全部未推送过的本地标签

git push prigin :refs/tags/v1.0//删除远程标签

//更换仓库

git remote -v

git remote rm origin

git remote add github git@github.com:18960908250/myGit.git

//同时关联githun与码云

//名字不一致就行

git remote add github git@github.com:18960908250/myGit.git

git remote add origin git@gitee.com:oriharaIZAYA/park.git

//搭建git服务器

Sudo apt-get install git

Sudo adduser git

Mkdir myGit

Cd myGit

Sudo git init –bare sample.git

Sudo chown –R git:git sample.git

修改/etc/passwd

Git:x:1001:1001:,,,:/home/git:/bin/bash替换成

Git::1001:1001:,,,:/home/git/usr/bin/git-shell

下载

Git clone git@server:/myGti/sample.git

git fetch origin master//从远程仓库更新

git config –global core.autocrlf false//禁用自动换行

git push –f origin master//强制覆盖

git pull origin master //等同于git fetch

git config branch.master.remote origin

git config branch.master.merge refs/hrads/master

//强制更新

git fetch --all

git reset –hard origin/master

git pull

git config –global color.ui true//使用配置颜色

<https://github.com/github/gitignore> //github自带的忽略配置文件

git check-ignore –v app.class //查看忽略规则

git config –global alias.st status //设置status的别名为st

cat .git/config //查看git配置文件

git rm –r –cached 1.txt //删除远程文件

git commit

git push

//更新失败后保留本地修改

git merge --abort

git reset --merge

git pull