#### 自我介绍

#### Git的简介及安装使用

1. 什么是Git：Git --- The stupid content tracker, 傻瓜内容跟踪器。Git是一个开源的分布式版本控制系统，可以有效、高速地处理从很小到非常大的项目版本管理。

远程项目仓库

(服务器仓库)

克隆

推送

…

本地

仓库1

本地

仓库2

本地

仓库n

分布式

1. Git的作用：
2. 主要：多人协作开发。团队合作共同开发一个项目，每个人负责不同的版块，完成后上传至服务器
3. 版本回退功能，随时切换到某一历史版本
4. 现场安装示例
5. 安装Git
6. 设置用户名邮箱以及查看

命令：git config --global user.name "Your Name"

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

git config user.name

git config user.email

1. **Git的工作原理和基础命令以及版本库简介**

添加

add

版本库(.git)

提交

commit

分支

工作区

（工作目录）

暂存区

1. 创建版本库

命令：git init 使用此命令多了一个隐藏文件.git

touch readme.txt (LICENSE git.c) 创建一个文件

git status 查看当前工作区状态

git add file1name file2name …

git status

git commit -m "注释"

git status

1. **Git指令操作之版本回退**

创建一个C语言程序

touch learning.c

git status

git add

git status

git commit -m

vim learning.c 修改 V0.1 从今天开始要认真学习，我一定能够学号Git

git diff

git status

git add

git status

git commit -m

git status

cat …

V0.2 沉迷学习,日渐消瘦，沉迷Git，无法自拔

V03 要什么女朋友，代码才是程序猿的女朋友

版本回退：

git log 查看修改日志，包含每一个版本的id

git log --pretty=oneline 简化信息

git reset --hard HEAD^ 回退到上一个版本，HEAD^^表示上上个版本

回退到某一个特定的版本

git reset --hard id id不需要完全

实质：修改HEAD指针指向的版本

git reflog 查看版本修改日志

小结：

HEAD指向的版本就是当前版本，因此，Git允许我们在版本的历史之间穿梭，使用命令git reset --hard commit\_id。

穿梭前，用git log可以查看提交历史，以便确定要回退到哪个版本。

要重返未来，用git reflog查看命令历史，以便确定要回到未来的哪个版本。

场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令

git checkout -- file。

场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令git reset HEAD file，就回到了场景1，第二步按场景1操作。

场景3：已经提交了不合适的修改到版本库时，想要撤销本次提交，参考版本回退一节，不过前提是没有推送到远程库。

删除文件：

本地文件：误删或者直接删除 rm filename

对于本地文件的删除恢复：git checkout -- filename 版本库里还有

版本库文件：git rm filename git commit -m “注释”删除

## 远程版本库

1. 创建本地电脑秘钥

ssh-keygen -t rsa -C "youremail@example.com"

一路回车即可，可以在用户主目录里找到.ssh目录，里面有id\_rsa和id\_rsa.pub两个文件，这两个就是SSH Key的秘钥对，id\_rsa是私钥，不能泄露出去，id\_rsa.pub是公钥，可以放心地告诉任何人

Git免费服务器： GitHub，SourceTree等 在上面创建一个账户

为什么GitHub需要SSH Key呢？因为GitHub需要识别出你推送的提交确实是你推送的，而不是别人冒充的，而Git支持SSH协议，所以，GitHub只要知道了你的公钥，就可以确认只有你自己才能推送。

当然，GitHub允许你添加多个Key。假定你有若干电脑，你一会儿在公司提交，一会儿在家里提交，只要把每台电脑的Key都添加到GitHub，就可以在每台电脑上往GitHub推送了。

最后友情提示，在GitHub上免费托管的Git仓库，任何人都可以看到喔（但只有你自己才能改）。所以，不要把敏感信息放进去。

1. 关联远程仓库
2. 添加公钥
3. 创建远程仓库
4. 关联 ： git remote add origin + 项目地址
5. 第一次推送： git push -u origin master
6. 更新：1. 添加修改到本地仓库

2. git push origin master

6）其它：git remote -v 远程版本库信息

1. 从远程仓库克隆

git clone + 项目地址

git pull 更新本地版本库

**作业：在GitHub中创建一个远程仓库并推送本地仓库的内容，本地仓库中至少有一个.txt文件写下自己学习Git的感受。提交方式: 周三晚上之前 发送一个word文档(记录做作业的过程)以及远程GitHub仓库地址至我的邮箱： huangyanlin@whut.edu.cn，有兴趣的话自学分支内容(可以添加到word文档中) 其它：遇到error 、困难、学会自主学习**