**Git学习笔记（一）**

1. Git和SVN的对比
   1. SVN

SVN是集中式版本控制系统，版本库是集中放在中央服务器的，而干活的时候，用的都是自己的电脑，所以首先要从中央服务器哪里的到最新的版本，然后干活，干完活，需要把自己做完的活推送到中央服务器。



但是集中式版本的控制工具缺点很明显：

* 服务器单点故障
* 容错性差
  1. Git

Git是分布式版本控制系统，那么它就没有中央服务器，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，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，那么多人如何写作呢？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，你们之间只需要把各自修改推送给对方，就可以互相看到对方的修改了。

图示

AI 生成的内容可能不正确。

1. Git的工作流程

一般的工作流程如下：（下载、修改、add添加到本地暂存、提交到本地仓库、push远程仓库用于共享）

1. 从远程仓库中克隆Git资源作为本地仓库；
2. 从本地仓库中checkout代码然后进行代码修改；
3. 在提交前先将代码提交到暂存区；
4. 提交修改，提交到本地仓库。本地仓库中保存修改的各个历史版本；
5. 在修改完成后，需要和团队成员共享代码时，可以将代码push到远程仓库；



1. Git的安装
2. Git的下载地址：<https://git-scm.com/downloads>
3. 安装Git for Windows(没有图形化界面)

表格

AI 生成的内容可能不正确。

1. 安装TortoiseGit（用于Git的图形化界面）

图片包含 表格

AI 生成的内容可能不正确。

1. 使用Git管理文件版本

版本库又名仓库，英语为repository。当我们开发一个程序时，需要用到Git，首先在文件夹中创建git。

1. 使用Git Bash Here，命令的方式实现

在该文件夹中点击右键中选择**Git Bash Here**来启动，图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

创建git版本库命令：git init

配置用户名/邮箱等：git config （git config –global user.name”名字”）

查看当前状态：git status

添加到暂存区：git add

提交修改：git commit

查看提交历史：git log

查看修改内容：git diff（查看前后两次提交版本的差别，修改了哪里）

查看分支：git branch

创建分支：git branch 新分支名

切换分支：git checkout 分支名

创建并切换：git checkout -b 分支名

合并分支到当前：git merge 分支名

删除本地分支：git branch -d 分支名

克隆远程仓库：git clone 地址名

查看远程地址：git remote -v

推送到远程仓库：git push

拉取远程最新代码：git pull

获取远程更新并不合并：git fecth

撤销对文件的修改：git checkout –文件名

回退到某个提交：git reset（git reset –hard 提交ID）

撤销一次提交（保留历史）：git revert（git revert 提交ID）

1. 使用图形化界面实现

添加：

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

提交：

图形用户界面, 应用程序

AI 生成的内容可能不正确。图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

查看历史：

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。

差异比较：

图形用户界面, 应用程序

AI 生成的内容可能不正确。

还原修改：当文件修改后不想把修改的内容提交，还想还原到未修改之前的状态。此时可以使用“还原”功能。（**注意：此操作会撤销所有未提交的修改，所以当作还原操作是需要慎重！**）

图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。

1. 远程仓库github

* 使用git bash here实现

在github上创建一个仓库，然后在要上传的文件夹中使用git bash here，并执行ssh-keygen -t rsa命令，然后根据生成的代码中找到生成的地址。

文本

AI 生成的内容可能不正确。

1. 密钥生成后需要在github上配置密钥本地才可以顺利访问。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

1. 同步到远程仓库，使用Git Bash Here,然后在git bash中执行如下语句：

origin是远程仓库的别名，可以是任意名称，

git remote add origin [git@github.com:bluecats2060/mytest.git](mailto:git@github.com:bluecats2060/mytest.git)

git push -u origin master

* 使用TortoiseGit实现

1. 由于TortoiseGit使用的ssh工具是“PuTTY”git Bash使用的ssh工具是“openSSH”，如果想让TortoiseGit也使用刚才生成的密钥可以做如下配置：

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

Url：远程仓库的地址

推送URL：也是相同的

Putty密钥：选择刚才生成的密钥中的私钥

然后在本地实现commit后，可以通过push提交到远程仓库

* 从远程仓库克隆到本地
  + 使用Git Bash Here实现

git clone [git@github.com:bluecats2060/mytest.git](mailto:git@github.com:bluecats2060/mytest.git)

文本

AI 生成的内容可能不正确。

* 使用TortoiseGit

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

* Git中从远程的分支获取最新的版本到本地有这样2个命令：
  + git fetch:相当于是从远程获取最新版本到本地，不会自动merge（合并代码）
  + git pull:相当于是从远程获取最新版本并merge到本地上述命令其实相当于git fetch 和git merge。在实际使用中，git fetch更安全些。

1. 分支管理

* 创建分支

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图形用户界面, 应用程序, 电子邮件

AI 生成的内容可能不正确。

* 切换分支

图形用户界面, 应用程序, Word

AI 生成的内容可能不正确。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

* 合并分支

我们在合并分支前，应该先将要合并分支里面的文件提交；

提交后再执行合并操作

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图形用户界面, 应用程序

AI 生成的内容可能不正确。