**Git入门**

1. **git概念**

Git（读音为/gɪt/）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。Git是目前世界上最先进的分布式版本控制系统，在处理各种项目时都十分高效，而且非常的高大上。

**2.git与svn的区别**

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得到最新的版本，然后干活，干完后，需要把自己做完的活推送到中央服务器。而且集中式版本控制系统是必须联网才能工作。

Git是分布式版本控制系统，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。

**3.Git安装及GitHub账号注册**

1.打开Git官网下载安装程序，然后按照默认选项安装即可。

安装完成后，打开Git bash软件，弹出一个类似cmd的命令行窗口，证明安装成功。

2.打开GitHub官网进行账号注册

1. **配置个人信息（名字与邮箱）**

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

$ git config --global user.email ["email@example.com"](mailto:\"email@example.com\")

查看是否配置成功，用命名git config -l

1. **创建本地仓库**

命令：git init（把当前目录变成一个git仓库）

以上命令会在当前目录下创建了一个.git 隐藏目录，它就是所谓的Git 仓库。生成仓库后的目录就不是普通的文档目录了，我们将其称为工作区，所以工作区中都包含一个git仓库，而一个git仓库中又包含一个暂存区和一个版本库。

查看仓库变更状态：git status

**6、工作区、暂存区、版本库概念**

Git有三大区 （ 工作区、暂存区、版本库 ）以及 几个状态 （ untracked、unstaged、uncommited ）

工作区（Working Directory）：电脑中的目录

仓库：工作区有一个隐藏目录.git，这个不算工作区，而是Git仓库。

暂存区（stage/index）：

版本库 （Repository） ：

**7、添加文件到版本库的步骤**

1）创建（修改）文件

往工作区中添加文件

2）添加到暂存区：git add <file>

把修改存放到一个临时的存储区域，Git 称之为暂存区（索引）。

3）提交到版本库：git commit -m "备注"

使用git commit 命令可将暂存区的内容提交至版本库中，这个过程称为提交，每一次提交都意味着版本在进行一次更新

**8、版本回退**

**1）显示从最近到最远的提交日志：git log**

1>--pretty=oneline（显示简要信息id+备注）

2>--graph（图形显示版本走向）

3>--abbrev-commit（显示简写的id）

4>一大串类似3628164...882e1e0的是commit id（版本号）

**2）当前版本：HEAD**

上一个版本：HEAD^

上上个版本：HEAD^^

...依此类推

前100个版本：HEAD~100

**3）git reset (--hard、--mixed、--soft)**

1>回退到上一个版本

git reset --hard HEAD^

2>回退到指定版本：

git reset --hard [commit id] 版本号没必要写全，前几位就可以了，Git会自动去找。

3>回退指定文件

git reset --hard [commit id] <file>

4>参数：(--hard、--mixed(默认值)、--soft)

--hard:工作区、暂存区、版本库的文件同时回退

--mixed：暂存区、版本库的文件回退（默认）

--soft：仅仅回退版本库中的文件

1. **查看命令历史：git reflog**

**9、远程仓库**

**1）创建SSH Key：**ssh-keygen -t rsa -C 'xxx@xx.com'

以上命令会在当前window用户的目录里创建.ssh目录，里面有id\_rsa（私钥）和id\_rsa.pub（公钥）两个文件

**2）把私钥或公钥添加到远程仓库**

**3）建立本地仓库与远程仓库的连接（先有本地仓库，后有远程仓库的时候，如何关联远程仓库）**

git remote add origin 远程仓库地址

PS：删除远程仓库连接：git remote remove <远程仓库名>

Ps:重新配置远程仓库地址：

         查看仓库是否配置成功：git remote -v

**4）从远程库克隆（先有远程库，后有本地仓库的时候，如何关联远程仓库）**

git clone 远程仓库地址

\*\*当你从远程仓库克隆时，实际上Git自动把本地的master分支和远程的master分支对应起来了，并且，远程仓库的默认名称是origin

\*\*测试线路是否连通：ssh -T [git@github.com](mailto:git@github.com)

**10、推送到远程仓库**

git push

格式：git push <远程主机名> <本地分支名>:<远程分支名>

\*\*第一次使用git push -u origin master后，之后就可以用简写命名：git push origin

**11、拉取与合并（同步本地与远程仓库）**

**1）git pull**

格式：git pull <远程主机名> <远程分支名>:<本地分支名>

--allow-unrelated-histories

**2）合并：** git merge       例：git merge dev 把dev分支合并到当前分支

\*\*push和pull后的分支顺序格式：<来源地>:<目的地>

**12、分支**

查看分支：git branch

添加分支：git branch  分支名

切换分支：git checkout  分支名

删除本地分支：

删除远程仓库分支：

合并分支： git merge  分支名       例：git merge dev 把dev分支合并到当前分支

特殊分支： gh-pages    展示自己项目