**面试题89答案**

[一 注册一个github账号，在自己的电脑上安装git？ 2](#_Toc13132)

[1 注册一个github账号 2](#_Toc4047)

[2 在自己的电脑上安装git 2](#_Toc18229)

[二 什么是版本控制系统？ 2](#_Toc4516)

[1 版本管理系统 2](#_Toc6687)

[2 git历史 3](#_Toc23622)

[3 GIT提供以下功能 3](#_Toc2617)

[三 Git是集中式还是分布式，二者的区别？ 3](#_Toc19864)

[1 集中式版本控制系统 3](#_Toc23683)

[2 分布式版本控制系统 3](#_Toc8863)

[四 Git的基本工作流？ 4](#_Toc15847)

[1 在工作区编辑代码 4](#_Toc4265)

[2 将工作区的修改添加到暂存区 4](#_Toc8773)

[3 将暂存区的内容提交到历史区 4](#_Toc10427)

[五 如何创建版本库，如何初始化目录？ 4](#_Toc6310)

[六 如何添加文件？ 4](#_Toc24431)

[七 如何提交文件？ 4](#_Toc9772)

[八 Git包含哪三个区域，如何查看状态？ 5](#_Toc9372)

[1 git包含三个区域 5](#_Toc1414)

[2 工作区有一个隐藏目录.git，这个不算工作区，而是Git的版本库。 5](#_Toc4491)

[3 如何查看三个区的状态差异 5](#_Toc22246)

[九 每个常用命令的作用？ 6](#_Toc10626)

[1 linux命令 6](#_Toc27859)

[2 git命令 6](#_Toc18407)

[3 切换远程仓库地址 7](#_Toc1091)

[4 上传文件 7](#_Toc28755)

[5 从远程服务器上拉取代码 7](#_Toc27370)

[6 从远程服务器上下载代码 8](#_Toc6208)

**一 注册一个github账号，在自己的电脑上安装git？**

**1 注册一个github账号**

http://jingyan.baidu.com/article/4e5b3e192200e291911e2468.html

**2 在自己的电脑上安装git**

http://jingyan.baidu.com/article/a501d80c22f0acec620f5e61.html

**二 什么是版本控制系统？**

**1 版本管理系统**

文件版本管理系统支持用户直观方便地进行文件存取、版本更新、历史版本浏览、文件版本比较、签出签入、版本回滚等操作。

**2 git历史**

2005年，git诞生。Linus花了 两周时间 自己用 C 写了一个分布式版本控制系统，这就是Git！一个月之内，Linux系统的源码已经由Git管理了！

2008年，github诞生。it迅速成为最流行的分布式版本控制系统，尤其是 2008 年 GitHub 网站上线了，它为开源项目 免费提供Git存储 ，无数开源项目开始迁移至GitHub，包括jQuery，PHP，Ruby等等。

**3 GIT提供以下功能**

3.1 备份文件

类似于用U盘网盘备份电影。我们的代码也需要备份。修改完了以后提交给版本库进行保管，哪一天代码没了也可以找回来。

3.2 记录历史

版本管理工具会记录是哪个人，在什么时候，修改了什么东西。

3.3 回到过去

如果我有一天不小心删除了某个文件，我们可以通过历史备份找回来

3.4 多端共享

手机 PAD 都可以共享代码

3.5 团队协作

多人协作，处理冲突

3.6 开源协作

查看别人的开源代码，开源分享自己的项目

**三 Git是集中式还是分布式，二者的区别？**

Git是分布式版本控制系统。

**1 集中式版本控制系统**

版本库是集中存放在 中央服务器 的，而干活的时候，用的都是自己的电脑，所以要先从中央服务器取得最新的版本，然后开始干活，干完活了，再把自己的活推送给中央服务器。

集中式版本控制系统的中央服务器要是出了问题，所有人都没法干活了。

集中式版本控制系统最大的缺点就是必须联网才能工作，同步代码到远程服务器受网速影响。

**2 分布式版本控制系统**

首先，分布式版本控制系统根本没有中央服务器，每个人的电脑上都是一个完整的版本库。

分布式版本控制系统的安全性要高很多，某一个人的电脑坏掉了不要紧，随便从其他人那里复制一个就可以了。

这样，工作的时候，就不需要联网了

**四 Git的基本工作流？**

**1 在工作区编辑代码**

工作区就是.git文件夹所在的目录，当前打开git bash命令行工具的文件夹。

**2 将工作区的修改添加到暂存区**

git add -A 提交所有修改

git add index.html 提交指定文件的修改

工作区和历史提交的中间缓存，代表需要提交的工作状态，临时保存你的改动。

比如git add 1.html

gti add 2.html

git add 3.html

git commit -m"提交是三个文件的改动"

之所以存在暂存区，为了避免多次提交导致历史区的记录太多，我们可以先将多次修改添加到暂存区，最后再一次性提交到历史区。

**3 将暂存区的内容提交到历史区**

git commit -m"提交所有文件“

git commit index.html =m"提交指定的文件"

历史区，历史仓库,指向你最近一次提交后的结果

注意：-m后面没有空格，双引号内可以是任意字符表示注释简要说明提交的内容。

**五 如何创建版本库，如何初始化目录？**

以window平台为例

1 安装git

2 在需要创建版本库的文件夹内右键打开git bash

3 在git bash命令行窗口输入git init命令后回车，则文件内多处一个.git文件夹则创建版本库成功。

初始化目录命令：git init

**六 如何添加文件？**

将工作区的修改添加到暂存区

git add -A 提交所有修改

git add index.html 提交指定文件的修改

**七 如何提交文件？**

将暂存区的内容提交到历史区

git commit -m"提交所有文件“

git commit index.html =m"提交指定的文件"

历史区，历史仓库,指向你最近一次提交后的结果

注意：-m后面没有空格，双引号内可以是任意字符表示注释简要说明提交的内容。

**八 Git包含哪三个区域，如何查看状态？**

**1 git包含三个区域**

1.1 工作区

在工作区编辑代码

工作区就是.git文件夹所在的目录，当前打开git bash命令行工具的文件夹。

注意：工作区不包括.git文件夹

1.2 暂存区

将工作区的修改添加到暂存区

git add -A 提交所有修改

git add index.html 提交指定文件的修改

工作区和历史提交的中间缓存，代表需要提交的工作状态，临时保存你的改动。

注意：.git文件中的index文件保存了暂存区域信息。

1.3 历史区

git commit -m"提交所有文件“

git commit index.html =m"提交指定的文件"

历史区，历史仓库,指向你最近一次提交后的结果

注意：-m后面没有空格，双引号内可以是任意字符表示注释简要说明提交的内容。

**2 工作区有一个隐藏目录.git，这个不算工作区，而是Git的版本库。**

.git文件夹下包含的子文件夹和文件说明：

2.1 description 文件仅供 GitWeb 程序使用，所以不用关心这些内容。

2.2 config 文件包含了项目特有的配置选项，

2.3 info 目录保存了一份不希望在 .gitignore 文件中管理的忽略模式 (ignored patterns) 的全局可执行文件。

2.4 hooks 目录包住了第六章详细介绍了的客户端或服务端钩子脚本。

另外还有四个重要的文件或目录：HEAD 及 index 文件，objects 及refs 目录。这些是 Git 的核心部分。

2.5 objects 目录存储所有数据内容，

2.6 refs 目录存储指向数据 (分支) 的提交对象的指针，

Git为我们自动创建的第一个分支master，以及指向master的一个指针叫HEAD。

2.7 HEAD 文件指向当前分支，

ref: refs/heads/master

默认指向master

2.8 index 文件保存了暂存区域信息。

其他文件夹或文件：

2.9 COMMIT\_EDITMSG 最后一次提交的commit注释内容

2.10 logs 记录commit历史记录的地方

**3 如何查看三个区的状态差异**

git status

git status命令可以列出所有在工作区与暂存区不同步，或者暂存区与历史区不同步的文件。

3.1 当工作区与暂存区不同步时

当我们在本地文件内修改了文件index.html，如果还没有执行git add =A命令，先执行git satus命令，则会出下列提示。

modified: index.html

注意：当本地文件发生变化，但没有执行add命令时，执行git status后最后一行提示文字信息为红色

3.2 当暂存区与历史区不同步时

当我们在本地文件内修改了文件index.html，已经执行了git add =A命令，但还没有执行git commit -m"提交"命令，则会出下列提示。

modified: index.html

注意：当本地文件发生变化，但只执行add还没有执行commit命令时，执行git status后最后一行提示文字信息为绿色

3.3 当工作区、暂存区和历史区同步时

三个区域同步，执行git status后出现下列提示表示工作区干净三者同步：

nothing to commit, working directory clean

注意：当本地文件没有变化，或者即使发生变化也已经执行过add和commit命令使得三个区域同步，然后再执行git status后出现的提示信息文字颜色统一为白色。

**九 每个常用命令的作用？**

**1 linux命令**

1.1 pwd 显示当前工作目录

1.2 ls -a 显示当前目录下的所有子文件夹和文件

1.3 cd /e/project 切换到指定目录

1.4 cd .. 返回上一级目录

1.5 cat cxh.txt 读取并在命令行窗口显示指定文件中的内容

1.6 echo 1 > cxh.txt 将cxh.txt文件中文字内容替换成1，覆盖原来的内容

1.7 echo 2 >> cxh.txt 在cxh.txt文件的末尾添加字符2，只在末尾补充追加依然保留原来的内容

1.8 mkdir test 在当前目录下创建文件夹test

**2 git命令**

2.1 全局配置git版本库用户邮箱(同一台电脑只需配置一次，写github注册邮箱)

git config --global user.email "handawnc@gmail.com"

2.2 全局配置git版本库用户名(同一台电脑只需配置一次，写github用户名)

git config --global user.name "caoxiaohan"

2.3 删除当前关联的远程仓库(如果想要添加新的仓库地址，应该先删除原来的)

remote rm origin

2.4 添加新的远程仓库地址(如果只操作同一个仓库地址，同一个文件夹下只需要执行一次)

git remote add origin https://github.com/caxiaohan/interview.git

2.5 初始化git版本(同一文件夹下只需要执行一次，如果出现.git文件夹就不要执行该命令)

git init

2.6 添加所有工作区的文件到暂存区

git add -A

2.7 提交所有暂存区的文件到历史区

git commit -m"上传"

2.8 推送文件到远程服务器

git push origin master

2.9 从远程服务器上拉取代马

git pull origin master

2.10 从远程服务器上下载代码

git clone https://github.com/caoxiaohan/interview.git

2.11 查看状态，检测三个区域是否一致

git status

2.12 查看提交日志，commit历史记录、commit注释信息及提交作者和邮箱

git log

**3 切换远程仓库地址**

第1步 删除当前关联的远程仓库

remote rm origin

第2步 添加新的远程仓库地址

git remote add origin https://github.com/caoxiaohan/interview.git

**4 上传文件**

第1步 当创建一个新文件夹或者打开不包含.git文件夹的普通文件夹时需要多执行下列两步：

git init

第2步 添加一个远程仓库，远程仓库的地址写自己的必须知道用户名和密码，或者拿到授权。

git remote add origin https://github.com/Kingziqiang/interview.git

第3步 添加所有文件

git add -A

第4步 提交所有文件

git commit -m"上传"

第5步 推送文件到远程服务器

git push origin master

分别输入github用户名和密码回车

**5 从远程服务器上拉取代码**

第1步 当创建一个新文件夹或者打开不包含.git文件夹的普通文件夹时需要多执行下列两步：

git init

第2步 添加一个远程仓库，可以是自己的也可以是别人的，不需要验证用户名密码。

git remote add origin https://github.com/caoxiaohan/interview.git

第3步 把远程服务上的代码拉到本地当前文件夹内

git pull origin master

**6 从远程服务器上下载代码**

第1步.把远程仓库克隆到本地

git clone https://github.com/caoxiaohan/interview.git

第2步.进入这个仓库

cd interview