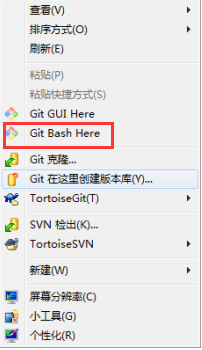
**Git的使用**

第一步

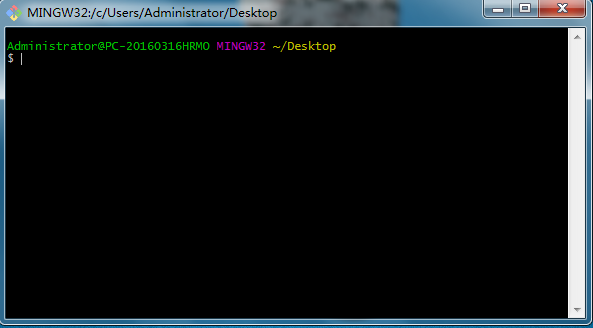
在桌面或者在D盘里面建立一个文件夹名字随意

第二步

右键单击新建的文件夹，左键点击红色标记处



第三步弹出命令框



命令模式（hard 意思是指针 . master当前分支 . status状态）

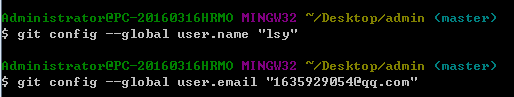
1. 首先输入 git init 意思是初始化版本库 输入完git init以后在文件夹里面会出 现 .git 不要动它（master代表的是主分支）（一般情况下 4个或者三个人一组，都用主 分支开发）

Git init admin 初始化admin本地库

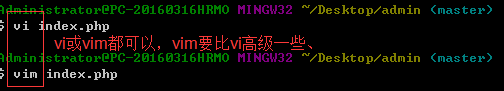
1. cd admin 切换到admin目录下

aaaa

1. 告诉git我是谁



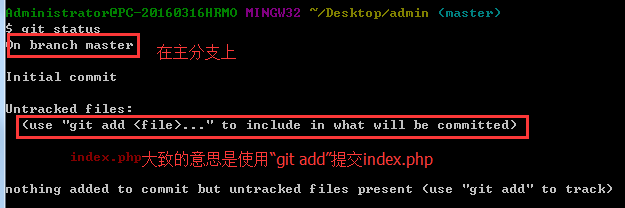
1. 在admin的文件夹中创建index.php，并写入内容



1. 查询admin目录下的内容



1. 查询其状态 git status



1. 添加index.php git add index.php



Git add --all将当前工作区所有内容添加

Git add . 将所有git中的修改文件添加

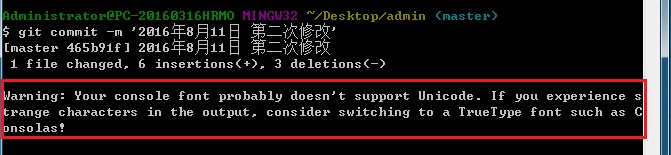
Git add （你要添加的文件名）精确添加

查看其状态：



8.git commit -m ’ 当前提交的注释说明 ’





意思：

您的控制台的字体可能不支持Unicode,如果你在输出中遇到奇怪的字符，考虑切换到一个TrueType 字体如Consolas 【 (Unicode:单一码 TrueType:微软和Apple公司共同研制的字型标准 Consolas:孔索拉)这三个在这里应该都是字体的意思】（仅供参考）

运行cmd → 上侧框兰，右键 → 属性 → 字体 → 字体下选择“非”点阵字体 → 回到git下 → git add index.php → git commit -m ‘正常写汉子即可’

出现：

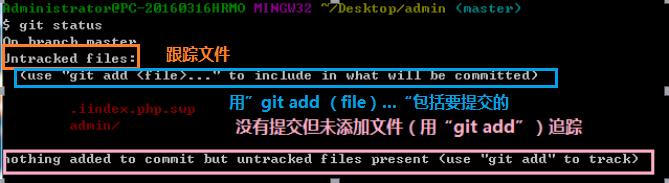


表示成功

1. 检查其状态



出现这样的原因是：你没有用git add 添加，直接用了git commit -m’’



1. 再次用vi或vim修改index.php后，查看修改后的不同之处

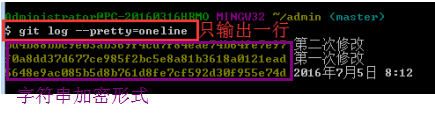
git diff index.php 查看修改后的不同之处



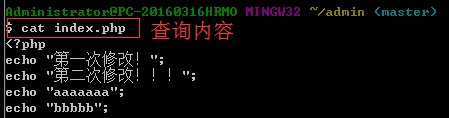
11.git log 查看历史版本



12.git log --pretty=oneline 一行显示



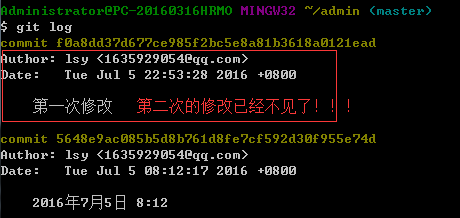
13.cat index.php 查询内容(cat Linux命令)



14.git reset --hard HEAD^ 回退到上一次版本号



15.git log 查看经过回退的历史



git reset --hard HEAD~10 回退上10次

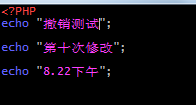
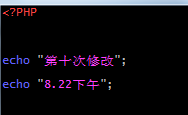
git reset --hard 2AA93B2(字母大小写都可以) 回退到指定版本



git reflog 查看版本行为（行为日志）

1. git checkout -- index.php 撤销index.php在工作区的修改

前： 后：

git reset HEAD index.php    将暂存区的文件撤销，重新放回工作区（但是修改的内容并不会删除，需要删除，还需要git checkout —index.php）

1. 删除文件

两种办法

     1.rm index.php     删除工作区内容

     2.git rm index.php 删除版本库

18.撤销办法：

    删除文件，还没有提交：直接使用git checkout — index.php

    已经提交，找回删除文件（版本回退）：使用 git reset —hard HEAD^

1. 将代码存储到远程仓库（Git支持的协议很多，最安全的方式：采用非对称密钥对）

1).生成密钥对

ssh-keygen -t rsa -C [“1635929054@qq.com”](mailto:\“1635929054@qq.com\”)

2).生成密钥

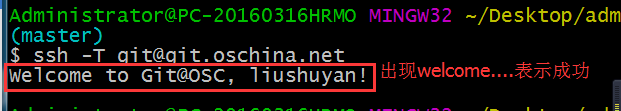
cat ~/.ssh/id\_rsa.pub

3).把生成的密钥粘贴在



4).代码验证密钥是否成功

ssh -T [git@git.oschina.net](mailto:git@git.oschina.net)



20.

git pull    更新服务器代码到本地

git pull --rebase origin master是将origin这个版本库的代码更新到本地的master主分支  
git push    将本地代码提交到服务器

git pull origin master --allow-unrelated-histories

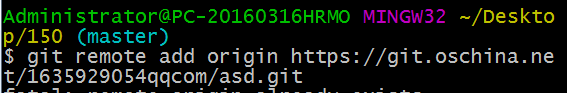
下载

21.现有本地库，再有远程库，做本地库关联远程库

     1.在OSchina上新建一个空仓库

     2.将本地库与远程仓库链关联，可以更新获取数据

git remote add origin + 你的远程地址



3.git remote -v 查看当前远程仓库的地址

git remote rm origin 删除远程地址

4. 如果





则必须用

如果创建时没有选择“README.md”则不用git pull origin master

git pull origin master (本地与远程合并)

     5.git push -u origin master   向远程仓库提交文件（推送）

.分布拉取

git fetch origin //将远端库获取本地但不合（适用于分支操作）

git merge origin //将本地的远端库合并

  先创建远程库，从远程库克隆（一般就是下载别人的源代码）

22.克隆

git clone <https://git.oschina.net/1635929054qqcom/bbs.git>

mkdir 文件名 //创建克隆地点

cd 文件名

git clone git@git.oschina.net:1635929054qqcom/bbs.git //git协议

或

git clone https://git.oschina.net/1635929054qqcom/bbs.gits //https协议

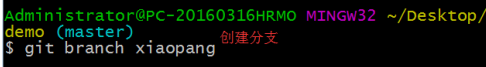
再看已将远端库bbs同步至本地clonedir目录。

23.创建与合并分支

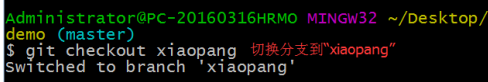
查看分支：git branch



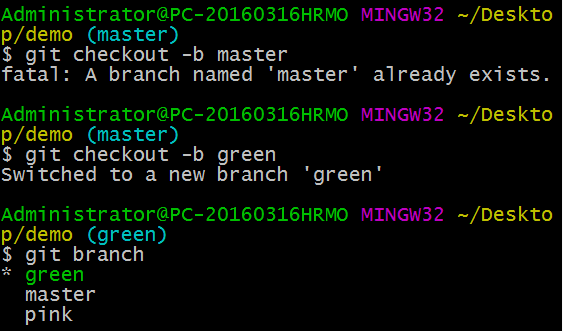
创建分支：git branch <name>



切换分支：git checkout <name>



创建+切换分支：git checkout -b <name>

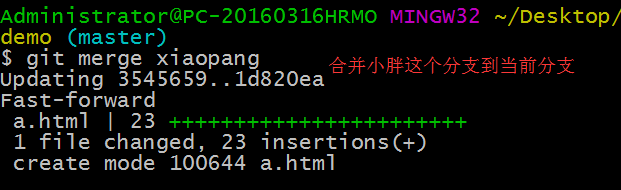


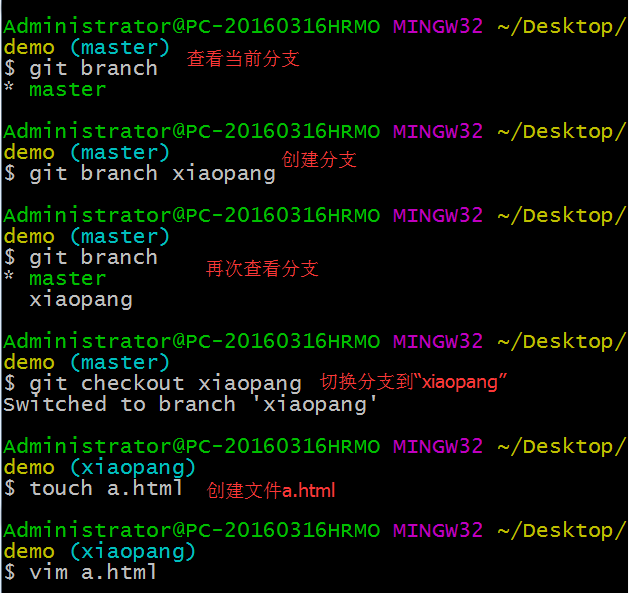
在新分支下创建文件 touch XXX.XX 并编辑添加文件

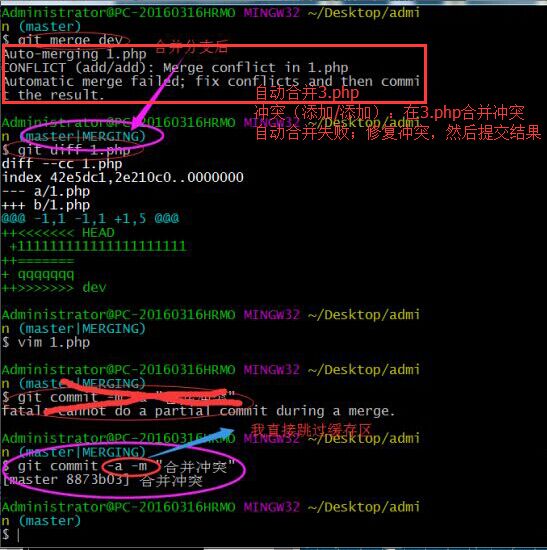
-

合并某分支到当前分支：git merge <name>

合并分支的情况必须是：文件名同名、同名文件内内 容内 在 同行内内容相同



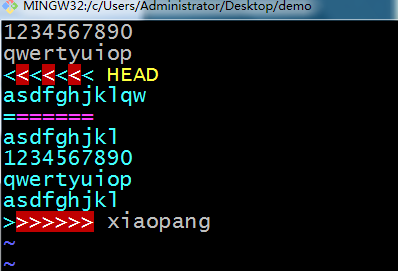




删除分支：git branch -d <name>

24.解决冲突：

     因为是直接修改文件，所以直接去掉不需要的内容



  查看分支的合并情况

git log --graph --pretty=oneline

git log --graph --pretty=oneline --abbrev-commit

  查看分支合并情况

  git log --graph

25.分支管理策略



图 5-7 分支管理策略

## 27.多人协作

将oschina/GitHub当做中心库协同开发，需要将本地文件推送至远端，远端拉取到本地两种操作类似SVN的提交/更新。

1. 查看远端库列表

git remote -v

1. 推送/拉取

git push origin branch //将本地库推送到远程banch分支

git pull origin branch //将远端banch分支拉取到本地并且合并

1. 分布拉取

git fetch origin //将远端库获取本地但不合并

git merge origin //将本地的远端库合并