```
git config --global user.email "zahuifan@163.com"
                                                                    git config --global color.ui "always"
                                                                                                    1 新增一个名为 GIT_EDITOR 的系统环境变量,设置它的值为 EmEditor.exe //sp。
                                                                    将 EmEditor 设置为 Git 的编辑器
                                                                                                    2 修改 PATH 系统环境变量,追加 F:\工具箱\文档工具\EmEditor

全局变量

                                                                                       解决 bash 控制台 ls 命令中文文件名乱码
                                                                                                                         在 Git\etc\git-completion.bash 文件中追加一行 alias Is='ls --show-control-chars --color=auto'
                                                                                                                                                      set output-meta on
                                                                                       解决 bash 控制台 git commit 无法输入中文注释
                                                                                                                              修改 Git\etc\inputrc文件
                                                                                                                                                      set convert-meta off
                                                                     解决中文乱码问题
                                                                                       解决 git log 命令中文注释乱码(只 bash 控制台好用)
                                                                                                                                  在 Git\etc\profile 中追加一行: export LESSCHARSET=iso 8859
                                                                                                                git config --global i18n.commitencoding ISO-8859
                                       初始化
                                                                                       解决 gitk 显示中文注释乱码
                                                                                                               git config --global i18n.logoutputencoding ISO-8859
                                                   初始化新版本库
                                                                                  只会在根目录下创建一个名为 .git 文件夹,非常清爽!
                                                                        git init
                                                                                                  1) 在根目录新建一个名为 .gitignore 的文本文件
                                                                                                                                         在命令行执行 echo *.jpg>.gitignore
                                                                                                                                                                         注意 > 号左右不要有空格
                                                                        设置每个人都想要忽略的文件
                                                    设置忽略的文件
                                                                                                  2 将 .gitignore 文件加入版本库并提交。
                                                                                                  修改 .git/info/exclude 文件
                                                                                                                           可使用正则表达式
                                                                                                                                             例如 *.[oa] 等价于 *.o 和 *.a
                                                                        设置只有自己需要忽略的文件
                                                                            添加单个文件
                                                                                           git add somefile.txt
                                                   添加新文件到版本库
                                                                            添加所有txt文件
                                                                                             git add *.txt
                                                                            添加所有文件
                                                                                           git add .
                                                                                                       ● 包括子目录,但不包括空目录
                                                    提交
                                                              git commit -m "add all txt files"
                                                                 提交所有修改
                                                                               git commit -m "some msg" -a
                                                                提交单个文件
                                                                                git commit -m "add msg to readme.txt" readme.txt
                                                                            git commit -C head -a --amend
                                                                                                         不会产生新的提交历史记录
                                                                 增补提交
                                                                                        撤销1、2个文件
                                                                                                         git checkout head readme.txt todo.txt
                                                                    撤销尚未提交的修改
                                                                                        撤销所有txt文件
                                                                                                         git checkout head *.txt
                                                                                        撤销所有文件
                                                                                                       git checkout head
                                                                                                               例:反转最近一次提交
                                                                                                                                                               相当于提交最近的一次提交的反操作
                                                      撤销修改
                                                                                             反转提交但不提交
                                                                                 反转提交
                                                                                                                                    git revert -- no-commit head
                                                                                                      git reset head 或 git reset head <filename>
                                                                                          取消暂存
                                                                    撤销提交
                                                                                 复位
                                                                                                                                         不会在版本库中留下痕迹。
                                                                                          复位到head之前的那个版本
                                                                                                                   git reset --hard head^^
                                                                                                                                         我使用的这个版本两个"^"相当于一个"^"
                                                                 列出本地分支
                                                                               git branch
                                                                 列出所有分支
                                                                               git branch -a
                                                                 基于当前分支的末梢创建新分支
                                                                                           git branch <branchname>
                                                                            git checkout <branchname>
                                                                 检出分支
                          Local
                                                                 基于当前分支的末梢创建新分支并检出分支

▼ git checkout -b <br/>
branchname>

                                                                                                git branch emputy bfe57de0
                                                                                                                           用来查看某个历史断面很方便
                                                                 基于某次提交、分支或标签创建新分支
                                                                                                git branch emputy2 emputy
                                                                                                                                如果发生了冲突,就不会自动提交,如果冲突很多,不想立即解决它们,可以直接使用 git checkout head . 撤销
                                                                                         合并并提交
                                                                                                       git merge <br/>branchname>
                                                      分支
                                                                             普通合并
                                                                                                                                🦞 实测结果 --no-commit 参数并未生效
                                                                                         合并但不提交
                                                                                                        git merge --no-commit
                                                                                         压合合并后直接提交
                                                                合并分支
                                                                                                            git merge -- squash < branchname>
                                                                             压合合并

√ git merge --squash --no-commit <branchname>

                                                                                         压合合并但不提交
                                                                                                                                                       当两个人合作开发一个新功能时,需要在一个分支上提交多次,开发完成之后再压合成一次提交
                                                                                                                                                   ₹ 但是要合并的提交只要比当前高2个版本,就会出现莫名其妙的冲突
                                                                                                                 git cherry-pick --no-commit 5b62b6
                                                                                         挑选某次提交合并但不提交
                                                                             拣选合并
                                                                                                                   不会覆盖已存在的同名分支
                                                                              git branch -m <branchname> <newname>
                                                                 重命名分支
                                       日常操作
                                                                              git branch -M <branchname> <newname>
                                                                                                                   会覆盖已存在的同名分支
                                                                                                 如果分支没有被合并会删除失败
                                                                             git branch -d new2
                                                                 删除分支
                                                                             git branch -D new2
                                                                                                 即使分支没有被合并也照删不误
                                                                    冲突很少时
                                                                                  直接编辑有冲突的文件然后提交即可
                                                                                                     1) 会生成.BACKUP, BASE, LOCAL和.REMOTE四个文件
                                                      解决冲突
                                                                                                     然后自动调用冲突解决工具,比较神奇的是如果安装了TortosieSVN 就会自动调用 TortosieMerge
                                                                    冲突比较负杂时
                                                                                     git merge tool
                                                                                                     3 解决之后手动删除 .orig 文件 (冲突解决之前的文件备份)
                                                                             为当前分支最近一次提交创建标签
                                                                                                         git tag 1.0
                                                                                                                       注意标签无法重命名
                                                                             为 Contacts 分支最近一次提交创建标签
                                                                 创建标签
                                                                                                                                          也可以把标签命名为 Contact/1.1
                                                                                                              git tag contacts_1.1 contacts
                                                                             为某次历史提交创建标签
                                                                                                   git tag 1.1 4e6861d5
                                                                 显示标签列表
                                                                               git tag
                                                      标签
                                                                                               查看标签断面的很方便的方法,但是不能提交
                                                                 检出标签
                                                                             git checkout 1.0
msysGit1.7.1
                                                                                 git branch b1.1 1.1
                                                                 由标签创建分支
                                                                                 git checkout -b b1.0 1.0
                                                                 删除标签
                                                                            git tag -d 1.0
                                                                    当前状态
                                                                                 git status
                                                                                 git log
                                                                                                   查看当前分支历史记录
                                                      查看状态
                                                                    历史记录
                                                                                                              查看某分支历史记录
                                                                                          gitk <branchname>
                                                                                                      查看所有分支
                                                                                          gitk --all
                                                                    每个分支最后的提交
                                                                                        git branch -v
                                                                              git archive --format=zip head>nb.zip
                                                      其它
                                                                 导出版本库
                                                                              git archive --format=zip --prefix=nb1.0/ head>nb.zip
                                                                       git clone <url>
                                                                                        例: git clone d:\git\nb
                                                                                                   remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
                                                       克隆版本库
                                                                                                   remote.origin.url=d:\usb
                                                                       克隆之后会自动添加 4 个 config
                                                                                                   branch.master.re mote=origin
                                                                                                   branch.master.merge=refs/heads/master
                                         初始化
                                                                                      git remote add <别名> <远程版本库的URL>
                                                                 添加远程版本库的别名。
                                                                                                                    remote.origin.url=d:\usb
                                                      别名
                                                                                       添加别名后会自动添加 2 个 config
                                                                                                                    remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
                                                                 删除远程库的别名和相关分支
                                                                                           git remote rm <别名>
                                                       创建一个无本地分支的库
                                                                                                     当需要一个公用的中央库时,非常适合把它建成 bare 库

√ git init –bare

                                                                    列出远程分支
                                                                                  git branch -r
                                                         分支
                                                                   删除远程库中已经不存在的分支
                                                                                               git remote prune origin
                                                                                                                                    orgin 是远程库的默认别名
                                                                                                                 例: git fetch origin
                                                                           获取但不合并
                                                                                          git fetch <远程版本库>
                                                                                                                 例: git fetch d:\git\nb
                                                                                                               等价于 git pull origin
                                          日常操作
                                                         从远程库获取
                                                                                                               需配置 branch.master.re mote=origin
                                                                                                   git pull
                                                                                                               和 branch.master.merge=refs/heads/master
                                                                           获取并合并到当前本地分支
                                                                                                               如果使用了 git clone 克隆远程库会自动添加这个配置
                                                                                                   git pull d:\git\nb master
                                                                                                                           由于不是从默认远程库获取,所以要指定获取哪个分支
                                                       推入远程库
                                                                         git push origin master
                                                                                                远程库的 master 不能是当前分支
                                                                                     🚺 启动 Git Bash 控制台 (git-bash.bat)
                                                                                                                                                  ~/.ssh 就是 C:\Users\jcl\.ssh
                                                                                                                         $ mkdir key backup
                                                                                    ② 如果以前生成过 SSH Key,要先备个份
                                                                                                                         $ cp id rsa* key backup
                                                                                                                                               提示 Enter pass phrase 时要输入密码
                          Remote
                                                                 生成 SSH Key
                                                                                                       $ ssh-keygen -t rsa -C "zahuifan@163.com"
                                                                                     3 生成 SSH Key
                                                                                                                                               生成的 SSH Key 存放在 C:\Users\jcl\.ssh
                                                                                                                       点击链接 Account Settings | SSH Public Keys | Add another public key,将 id_rsa.pub
文件中的内容粘贴到 key 文本框中
                                                                                    4 将 SSH Key 添加到 GitHub 账户里
                                                                                                                                   输入密码后如果显示信息里包含 You've successfully authenticated 表示连接成功。之后会在
                                                                                    5 第一次连接到 GitHub
                                                                                                            $ ssh git@github.com
                                                                                                                                   C:\Users\jcl\.ssh 建立一个名为 known_hosts 的文件
                                                                 创建新库
                                                                                进入 Dashboard Tab页,点击 New Repository 按钮
                                                                                                                           免费的只能 Public!
                                                                  克隆版本库
                                                                                  进入新建的库,点击 SSH 按钮,复制版本库地址,然后使用 git clone 命令即可
                                                                                                                                              点击 Git Read-Only 按钮可获取只读权限的地址,克隆这个地址不需要密码
                                                                                                  1 把密码加密保存在硬盘上
                                                                                                                           $ ssh-keygen -p
                                                                                                                                          在 Git \etc\profile 中追加:
                                                                                                                                          # lanch ssh agent
                                         http://github.com/
                                                                                                                                          SSH_ENV="$HOME/.ssh/environment"
                                                                                                                                          function start_agent {
                                                                                                                                           echo "Initializing new SSH agent..."
                                                                                                                                           /usr/bin/ssh-agent | sed 's/^echo/#echo/' > "${SSH ENV}"
                                                                                                                                           echo succeeded
                                                                                                                                           chmod 600 "${SSH_ENV}"
                                                                                                                                            . "${SSH ENV}" > /dev/null
                                                                                                                                           /usr/bin/ssh-add;
                                                                                                  2 修改配置文件,让 ssh-agent 随 git bash 启动
                                                                 让 ssh-agent 替我们记住密码
                                                                                                                                          # Source SSH settings, if applicable
                                                                                                                                          if [ -f "${SSH_ENV}" ]; then
                                                                                                                                            "${SSH_ENV}" > /dev/null
                                                                                                                                           #ps ${SSH_AGENT_PID} doesn't work under cywgin
                                                                                                                                           ps -ef | grep ${SSH_AGENT_PID} | grep ssh-agent$ > /dev/null || {
                                                                                                                                            start_agent;
                                                                                                                                                                                      @Martin_Ye
                                                                                                                                          else
                                                                                                                                           start_agent;
                                                                                                                                                       weibo.com/209097982
```

3 重启 Git Bash 控制台

git config --global user. name "zahuifan"