**Git初始化**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

配置用户名和邮件地址

git config --global user.name "xxx"

git config --global user.email xxx@gmail.com

在git命令输出中开启颜色显示

git config --global color.ui true

新建工作目录，执行git init创建版本库

git init

有文件新建后执行

git add xx

如果是文件修改

git add

提交文件

git commit -m "xxxx"

其他命令

git log --stat

git log --pretty=oneling 精简输出来显示日志

git status -s 精简格式的状态输出

git diff /git diff HEAD / git diff --cached

Git 使用.gitignore文件来忽略不愿意提交的目录或者文件

然后就直接写要忽略的文件或者文件夹，例如忽略到cache目录

/cache

这样即可

git pull = git fetch + git merge

git pull实际包含两个动作： 获取远程版本库的最新提交，以及将获取到的远程版本库提交与本地提交进行合并。

git push 推送到远程版本库

git stash 保存进度 保存当前的工作进度。会分别对暂存区和工作区的状态进行保存。

git stash apply 恢复进度

git stash list 显示进度列表。此命令显然暗示了git stash 可以多次保存工作进度，并用在恢复时候选择。

本地删除如果要反映到暂存区中应该用git rm命令，对不想删除的文件执行git checkout --<file> 可以让文件在工作区重现。

git add -u 快速标志删除 将本地有改动（包括修改和删除）的文件标记到暂存区。

**Git 冲突解决**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**A:**

git add -u

git commit -m "xxxx"

git push

**B:**

git add -u

git commit -m "xxx"

git pull

提示冲突：

查看暂存区中有关冲突文件的多个不同版本的记录：

git ls-files 命令查看

编号为1 的暂存区用于保存冲突文件修改之前的副本

编号为2 的暂存区用于保存当前冲突文件在当前分支中修改的副本

编号为3 的暂存区用于保存当前冲突文件在合并中修改的副本

手动修改后， git add -u

git commit -m "xxxxx"

通过git log --oneline --graph-3 可以查看

**图形界面解决冲突：**

git mergetool

**Git 分支**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**