**LINUX命令**

pwd 显示当前目录

ls -ah 显示隐藏目录

cat <file> 显示某个文件内容

echo "hello git"

> index.html //将'hello git' 写入到index.html中 单个>箭头表示写入， >>表示追加

rm filename 删除文件

**GIT命令**

git init 创建版本库，初始化一个git仓库，会自动创建唯一的master分支

添加文件到git仓库

git add <files> ，可反复多次添加文件，实际上将文件修改添加到暂存区

git commit -m "本次提交的说明" ，实际上将暂存区所有内容提交到当前分支

git status 掌握工作区状态，查看是否有文件被修改

git diff <file> 查看修改的内容，比较暂存区与工作区的差异

git diff –cached 比较暂存区与历史区的差异

修改文件到git仓库与添加同，add-commit

git log 查看历史记录，显示由最近到最远的提交日志

git log --pretty=oneline 日志用一行显示

git中，HEAD表当前版本，HEAD^表上一个版本，HEAD^^上上个，HEAD~100表前100

git reset --hard HEAD^ 回退到上一个版本

git reset –hard 版本号 回退到某个版本，版本号可以只写前几位

git reflog 查看命令历史，可以确定要回退到未来哪个版本

本地git仓库与github传输，需要设置SSH加密

<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/001374385852170d9c7adf13c30429b9660d0eb689dd43a000>

关联本地库到远程仓库，现在github上创建repo

git remote add origin <https://github.com/xieyiyu/demo1.git>

将本地库内容推送到远程

git push –u origin master 第一次要加参数-u，之后在提交可不加

git clone 克隆远程库，通过ssh支持原生git协议速度快，use SSH地址

**GIT分支管理**

git branch 列出所有分支

git branch dev 创建分支

git branch –d dev 删除分支

git checkout dev 切换到dev分支

git checkout -b dev 创建并切换到dev分支

git merge dev 合并分支，先切换到master分支中

git无法自动合并分支时，需要先解决冲突在提交，git log –graph 查看分支合并图

git merge --no--ff -m “描述” dev

参数--no—ff表禁用fast forward 合并，该合并会创建一个新的commit

**Bug分支**

用git stash 存储当前工作现场

<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/00137602359178794d966923e5c4134bc8bf98dfb03aea3000>

**Feature分支**

添加新功能

<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/001376026233004c47f22a16d1f4fa289ce45f14bbc8f11000>

origin 远程仓库的默认名称

git remote 查看远程库信息 参数-v显示详细信息

master分支是主分支，因此要时刻与远程同步；

dev分支是开发分支，团队所有成员都需要在上面工作，所以也需要与远程同步；

bug分支只用于在本地修复bug，就没必要推到远程了，除非老板要看看你每周到底修复了几个bug；

feature分支是否推到远程，取决于你是否和你的小伙伴合作在上面开发。

clone默认只能看到master分支

git checkout –b dev origin/dev ，可创建远程origin的dev分支到本地

**多人协作**

多人协作的工作模式通常是这样：

首先，可以试图用git push origin branch-name推送自己的修改；

如果推送失败，则因为远程分支比你的本地更新，需要先用git pull试图合并；

如果合并有冲突，则解决冲突，并在本地提交；

没有冲突或者解决掉冲突后，再用git push origin branch-name推送就能成功！

如果git pull提示“no tracking information”，则说明本地分支和远程分支的链接关系没有创建，用命令git branch --set-upstream branch-name origin/branch-name。

**标签管理**

git tag 查看所有标签

git tag v1.0 在某个branch下打标签

git tag v0.9 123456（某个commit id）

git show v1.0 查看标签信息

git tag -a <tagname> -m "blablabla..."可以指定标签信息；

git tag -s <tagname> -m "blablabla..."可以用PGP签名标签；

git push origin <tagname>可以推送一个本地标签；

git push origin --tags可以推送全部未推送过的本地标签；

git tag -d <tagname>可以删除一个本地标签；

git push origin :refs/tags/<tagname>可以删除一个远程标签。

git init

git add .

git commit –m “”

git remote add origin XXX

git pull origin master

git push –u origin master

问题

git pull失败 : refusing to merge unrelated histories

应用：git pull origin master --allow-unrelated-histories 合并两个不同的项目

**Hexo命令**

hexo s --debug 启动本地站点并开启调试模式

hexo new page xxx 新建页面