Git

自定义git

- 忽略特殊文件
  - 例如一些隐私文件等不方便推送到远程库公开
    - 在工作区创建.gitignore文件
      - 在git status时会提示提交，强迫症难受
      - 提交gitignore文件，再次git status发现工作区清空，将工作区文件上传到github会忽略password.in文件
      - 远程库中的文件即github文件不删库只删除单个的文件，可以本地删除对应文件，然后将所有再次提交更新即可git add *
- 配置文件简化命令
  - 每个仓库的配置文件都放在.git/config文件中
    - git config --global alias.st status
    - git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit"
- 搭建git服务器

使用gitee

- 初始关联时发现远程库origin已经关联github
  - 方法一：移除github关联，关联gitee
    - 查看关联git remote -v
    - git remote rm origin
  - 第二种通过设置关联标识关联多个远程库
    - 同github一样设置ssh公钥设置仓库，进行关联，命令和geihub一样
    - git remote add github git@github.com:OceanStarT/ocean.git
    - git remote add gitee git@gitee.com:OceanStarT/ocean.git

使用Github

- 可以fork别人的项目，然后克隆到自己的本地上

标签管理

- 创建标签
  - 标签与commit id相对应
  - 进入需要打标签的分支
    - git tag v1.0
    - git tag查看标签
  - 也可以对某个commit id号打标签
    - git tag v1.1 d2ae4
  - 对标签进行说明
    - git tag -a v1.2 -m "version 1.2 released" d2ae4
    - 查看标签信息 git show v1.2
- 操作标签
  - 若标签错误进行删除,本地删除 git tag -d v1.1
  - 将标签推送到远程 单个标签 git push origin v1.0
  - 一次性推送所有未推送的标签 git push origin --tags
- 删除远程标签
  - 先本地删除，以v1.2为例 git tag -d v1.2
  - 在远程删除 git push origin :refs/tags/v1.2

分支管理

- 创建合并分支
  - 在ocean文件中添加内容显示 相当于创建并切换到dev分支 git checkout -b dev
  - 删除dev分支git branch -d dev git checkout dev
  - git merge dev合并分支 git branch dev
  - git switch master
  - git switch -c dev
- 解决冲突
  - 当在不同的分支上修改同一句话时，修改的不一样，合并就会出问题，此时可以查看内容，修改为一样在合并 git log --graph --pretty=oneline --abbrec-commit
- 分支管理策略
  - 合并分支时，普通合并时git使用Fast forward模式会丢失分支信息，因此可以用命令 git merge --no-ff -m 'merge with no-ff' dev
- Bug分支
  - 做项目中临时需要解决其他bug，所以当前分支保存 git stash
    - 恢复方法一 git stash pop恢复的同时删除stash内容
    - 恢复方法二
      - git stash apply
      - git stash drop
  - 若多次恢复的时候
    - git stash list
    - git stash apply stash@{0}
  - 若项目中相同的bug同时需要在另一个分支中提交 在dev分支下，git cherry--pick 4c805e2,会在当前分支下自动做一次提交
- Feature分支
  - 项目中添加新功能时，最好新建一个分支，但是最后未和主分支合并需要删除时 git branch -D feature-vulcan
- 多人协作
  - 从远程库克隆项目
    - git clone git@github.com:OceanStarT/ocean.git
    - 创建远程origin的dev分支到本地 git checkout -b dev origin/dev
- Rebase
  - git rebase

git简介

- 分布式版本控制系统
- git config --global user.name "name" git config user.name
- git config --global user.email "email" git config user.email
- 转到预设的目录下，如cd F:/git/ocean 进行 git init 设置仓库（ls -ah命令可以查看隐藏内容）
  - 注意，可一次提交多个文件

版本操作

- 版本回退
  - git status 注意：每次修改想提交的话必须先add在commit
  - git log(git log --pretty=oneline)查看修改的版本信息
  - git reset --hard head^(一个^为回退一次，每次回退head会被更新，若回退3次可以是^^^也可以是~3)
  - git reflog 查看所有版本号方便找回最先的版本然后使用git reset --hard ca48f设置最先版本为head
- 工作区与暂存区
  - 工作区->暂存区->版本库
  - 暂存区：git add test.txt
    - git add就是先把文件提交到暂存区，最后统一提交到版本库中
  - 版本库：git commit -m "add file"
- 管理修改 git管理的是修改，即每次工作区中的文件修改后没有进行add则无法进行commit
- 撤销修改
  - 文件在工作区 git checkout -- test.txt 注意"--"十分重要
  - 文件在暂存区 先退回到工作区 git reset head test.txt在checkout修改
  - 文件在版本库但还未提交到远程库 head找到想要的版本例如git reset --hard head^
- 删除文件
  - git rm test2.txt
  - git commit -m "remove test2.txt" 文件删除后也可以通过版本库中的版本号找回

远程仓库

- 账户链接
  - 在账户主目录下创建.ssh文件，里面会有id_rsa私钥和id_rsa.pub公钥文件,将公钥内容添加到github账户setting里面的 SSH keys里面
  - ssh-keygen -t rsa -C "youremail@email.com"
  - ssh -T git@github.com 验证链接
- 添加远程库
  - 在github网站里创建新的仓库
  - git remote add origin git@github.com:OceanStarT/ocean.git 在本地仓库下与其进行关联
  - git push -u origin master
  - 之后更新版本只需要git push origin master
- 远程库克隆
  - 移除本地仓库，回到上一级目录 rm -rf ocean/
  - git clone git@github.com:OceanStarT/ocean.git
  - touch ocean.txt echo happy > ocean.txt git add . git commit -m 'ocean txt'