git学习

用途：版本管理

存储：集中式和分布式 GIT属于后者

分布式：本地和服务器都有

安装

配置作者信息：

git config --global user.email “2353135.qq.com”

git config --global user.name “awei”

创建新仓库和维护旧仓库

git init : 初始化仓库

git clone: 克隆命令

流水线操作分析：

先将文件放在推车中 add，然后放在仓库 commit

Git status：查看状态

使用命令行实现流水线操作：

Git status: 查看生产车间状态，即init之后的仓库状态，可以看到哪些没有提交之类的

Git add 文件名

Git commit -m ‘注释’

Git add .: 可以全部入车

详解控制版本库文件管理

Git add . 可以将没添加的直接添加，但是如果有的不想提交呢

创建.gitgnore文件 忽略文件配置，

\*.txt 忽略所有.txt文件

a.txt 不添加a文件

\*.txt ！a.txt 忽略所有txt文件 除了a

/文件夹名：不添加该文件夹

/vendor/\*.php

/adsf/\*\*/index.php 文件夹的子文件的文件

从版本库中删除文件

Git rm 文件名 本地也丢了

Git rm --cached 文件名 本地不丢

版本库修改资源名字

Git mv 原文件名 新文件名 然后需要add commit

使用log日志看历史操作

Git log (-p) 详细 -1 只看最近一次 --name-only --oneline --name-status

使用ament修改最新一次提交事件

Git commit --amend

管理暂存区中的文件

Git rm --cached 文件 第一次将文件从add中撤出来

Git reset HEAD 文件 第二次以后撤销

Git checkout -- 文件 撤销到仓库上次保持的文件内容 即还原文件

Alias命令别名

Git config --global alias.a add 将add别名为a, 可git a .

Git分支Branch存在意义

创建分支 切换分支 删除分支 合并分支

实例讲解分支基本管理操作

Git branch 查看分支

Git branch 分支名

Git checkout 分支名 切换当前分支

Git checkout -b 分支名 创建并切换到该分支

分支的合并与删除

Git merge 分支名

Git branch -d 分支名 删除分支 合并后分支就可以删除了

分支冲突实例讲解

多个分支修改了同一个文件后，分支合并时会有冲突

分支管理merged及分支强制删除

Git branch --merged 已经合并的分支

Git branch --no-merged 没合并的分支

Git branch -D 分支名 强制删除分支

标志的分支工作流

创建默认的分支 master 稳定分支，新建一个分支develop，再新建分支

Stash临时储存区实例讲解

Git stash 将当前分支缓存 可切换到其他分支

Git stash apply 返回分支

Git stash list 列出缓存分支

Git stash drop 缓存区名

Git stash pop 恢复并删除

使用TAG标签声明项目阶段版本

Git tag 稳定的代码才有必要打标签 某个阶段的总结

生成zip代码发布压缩包

Git archive master --prefix=’文件’ --forma=zip > 压缩包名字

使用系统别名定义git全局指令

.bash\_profile文件

alias gs=”git status”

合并分支产生的实际问题演示

Log会有分支记录

Rebase合理的优化分支合并

将子分支移到主分支的最新点

Git rebase master (在子分支)

国内和国外项目托管平台介绍

github

SSH和github远程无密码连接

生成密钥 ssh-keygen -t rsa 一直回车

头像 设置 ssh 添加

Git push 将项目推到远程

有可能连不上 找到known\_hosts 删除内容

本地版本库使用remote与GitHub相关联

创建本地库并完成初始提交

添加远程仓库 git remote add origin [git@github.com:dasf/hd-xj.git](mailto:git@github.com:dasf/hd-xj.git)

查看远程库 git remote -v

推送数据到远程仓库 git push -u origin master

(通过clone的已经关联 不需要再次关联)

本地分支与github远程分支同步

Git branch -a 查看分支

Git push --set-upstream origin ask

新入职员工参与项目开发时分支使用

Git clone 项目 可能是为了建立连接

Git pull origin ask:ask 将分支拉下来

Git push --set-upstream origin ask

Github远程分支合并

和之前一样 需要pull

远程分支删除操作

Git push origin --delete ask

自动部署之流程分析与创建web站点

项目设置 点击webhooks 填写请求url

Github自动部署到web服务器

完结！！！