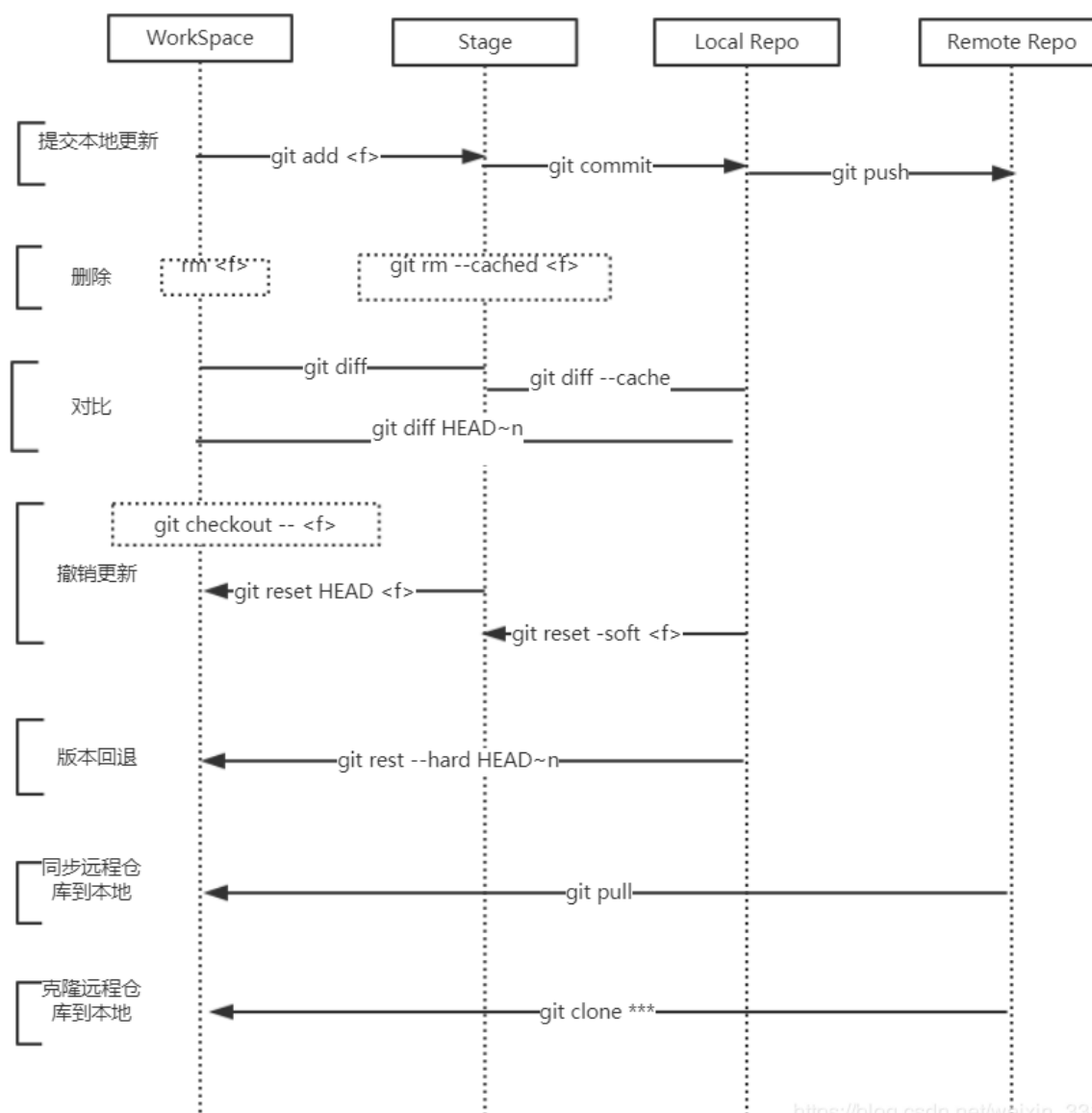


https://blog.csdn.net/wisixin_33127753



https://blog.csdn.net/wisixin_33127753

- 1 创建版本库
- 2 `git clone <url>` #克隆远程版本库
- 3 `git init` #在本地的当前目录里初始化git仓库
- 4

```
5 修改和提交
6 git status #查看WorkSpace的状态
7 git diff #显示WorkSpace和Stage中的状态差异
8 git add <file> / git add . #从WorkSpace保存到Stage,add后的文件才会被git跟踪
9 git mv <old> <new> #文件改名
10 git rm <file> #删除文件
11 git rm --cached <file> #从Stage中移除,停止跟踪文件但不删除
12 git commit -m "message" #从Stage提交更新到Local Repo
13 git commit --amend #修改最后一次提交
14
15 查看提交历史
16 git log #查看commit的历史记录(当前分支)
17 git log -p <file> #查看指定文件的提交历史
18 git blame <file> #以列表方式查看指定文件的提交历史
19 git log --pretty=format:"%h - %an, %ar : %s" 格式化查看(最优)
20
21 撤销
22 git checkout --<file> #撤销WorkSpace中的更新,将Stage的文件提取覆盖当前文件(撤销后无法找回)
23 git reset HEAD <file> #撤销Stage中的更新,移出到WorkSpace中(用于反悔 git add <file>)
24 #撤销Local Repo中的更新
25 #回退到相应版本号,同样也可以回退到未来的版本号:
26 git reset --hard HEAD^ 回退到上个版本
27 git reset --hard HEAD~3(n) 回退到前3次提交之前,以此类推,回退到n次提交之前
28 git reset --hard commit_id 退到/进到,指定commit的哈希码(这次提交之前或之后的提交都会回滚)
29
30 回滚后提交可能会失败,必须强制提交
31 强推到远程:(可能需要解决对应分支的保护状态)
32 git push origin HEAD --force
33
34 git reflog ##查看HEAD的变更记录,包括已经撤销的更新
35 #恢复撤销操作:
36 git reset --hard HEAD@{1}
37 git reset --hard d03ab23
38
39 #--head: 撤销并删除相应的更新
40 #--soft: 撤销相应的更新,把这些更新的内容放到Stage中
41
42 分支与标签
43 git branch #显示所有本地分支
44 git checkout <branch/tag> #切换到指定分支或标签
```

```
45 git branch <new-branch> #创建新分支
46 git branch -d <branch> #删除本地分支
47 git tag #列出所有本地标签
48 git tag <tagname> #基于最新提交创建标签
49 git tag -d <tagname> #删除标签
50
51 合并与衍合
52 git merge <branch> #合并指定分支到当前分支
53 git rebase <branch> #衍合指定分支到当前分支
54
55 远程仓库
56 git remote -v #查看远程版本库信息
57 git remote show <remote> #查看指定远程版本库信息
58 #添加远程版本库,把本地库和远程库关联起来
59 git remote add <remote> <url>
60 git remote add origin git@github.com:~
61 #从远程库获取代码
62 git fetch <remote>
63 #提交本地仓库到远程仓库
64 git push -u origin master #上传代码并快速合并
65 git push -u origin master -f #强制push, 不管冲突
66 git push -u origin master --allow-unrelated-histories #把两个不同的项目合并
67 git push <remote> :<branch/tag-name> #删除远程分支或标签
68 git push --tags #上传所有标签
69 #同步远程仓库到本地
70 git pull <remote> <branch> #下载代码并快速合并
71 git pull origin master
72
73 其他
74 #删除文件
75 rm <file>
76 git rm <file>
77
78 #创建分支, 提交到分支
79 git branch [name]
80 git push -u origin [name]
81
82 #删除当前目录中所有未追踪的文件
83 git clean -xf
84
85 #处理中文文件名
86 git config --global core.quotePath false
```

```
1 git init
2 初始化git环境，这样可以配置一些参数之类的。
3 git clone之后一般就自动初始化了
4
5 # 文件太大，添加压缩比
6 # 为 git 添加配置项，通过下面的命令可以简单完成
7 # 在这之前可以执行 git config -l 命令看看已有配置项有哪些
8 git config --add core.compression -1
9
10
11 git 密码输错后fatal
12 控制面板->用户账户->凭据管理器->Windows凭据
13 找到gitlab的
14
15
16 克隆到本地
17 git clone https://
18 如果出现fatal: protocol 'https' is not supported
19 是因为前面多一个空格，编码不对，需要删掉重新打一次就可以了
20
21 切换分支
22 git checkout zhengqi
23 查看状态
24 git status
25
26 git config --global core.autocrlf false
27
28 git add .
29
30 git commit -m 'test'
31
32 git push
33
34 删除远程提交地址
35 git remote rm origin
36
37 切换远程提交地址
38 git remote add origin https://github.com/scorpiozq/video_cut.git
39
40 git pull
1 git init
```

```
2 初始化git环境，这样可以配置一些参数之类的。
3 git clone之后一般就自动初始化了
4
5 # 文件太大，添加压缩比
6 # 为 git 添加配置项，通过下面的命令可以简单完成
7 # 在这之前可以执行 git config -l 命令看看已有配置项有哪些
8 git config --add core.compression -1
9
10
11 git 密码输错后fatal
12 控制面板->用户账户->凭据管理器->Windows凭据
13 找到gitlab的
14
15
16 克隆到本地
17 git clone https://
18 如果出现fatal: protocol 'https' is not supported
19 是因为前面多一个空格，编码不对，需要删掉重新打一次就可以了
20
21 切换分支
22 git checkout zhengqi
23 查看状态
24 git status
25
26 git config --global core.autocrlf false
27
28 git add .
29
30 git commit -m 'test'
31
32 git push
33
34 删除远程提交地址
35 git remote rm origin
36
37 切换远程提交地址
38 git remote add origin https://github.com/scorpiozq/video_cut.git
39
40 git pull
```