

Git基础

git创建分支

- git brach dev
创建一个名为dev的分支。
- git switch dev
表示切换分支到dev。
- git checkout dev, 也可以切换分支到dev。
- git brach
查看当前分支。
git brach会列出所有分支, 但会在当前分支前面加上1个"*"。
- git merge dev
表示合并指定分支到当前分支。
- git brach -d dev
表示删除指定分支。
- git分支的实质
git创建1个新的分支, 实际上是创建了1个新的指针, git brach dev创建了1个新的指针dev, 这个指针指向当前提交。
- git切换分支的实质
git切换分支实际上是将HEAD指针指向指定分支, 这样再做提交就在新分支上进行。
- 在git看来, 所有提交都是在一条时间线上, 当切换到新的分支, 再做提交, 就只有新分支的指针dev继续指向新提交, 原分支master指针不再移动。
- 分支的指针, 如master等是指向提交的, 而HEAD指针是指向当前分支的指针的, 切换分支实际上是改变了HEAD指针的指向。

删除文件

- 如果想要删除readme.txt, 我们可以直接在git bash输入rm readme.txt。(这样在工作区, readme.txt就被删除了)。
- 如果我们在版本库中也删除readme.txt, 我们可以使用git rm readme.txt将删除修改提交到暂存区, 再使用git commit -m "readme.txt"将删除操作提交到版本库。
- 如果我们想撤销这个删除, 可以参照撤销工作区的修改。

撤销在工作区的修改 (工作区是可以项目目录)

- git checkout -- readme.txt (可以撤销在工作区的修改)。
- git checkout -- readme.txt实际上是使用版本库中的版本替换工作区中的版本。
- 当修改已经add到暂存区, 那么使用git checkout -- readme.txt就不起作用了。
- git reset HEAD readme.txt可以撤销在暂存区的修改。
- 再使用git checkout -- readme.txt就可以撤销修改了。

工作区和暂存区

- 工作区就是在电脑里看到的目录就是工作区, 如learn git文件夹就是一个工作区。
- 工作区存在一个.git文件夹, 这个就是git的版本库, git版本库包含了很多东西, 其中最重要的是称为stage的暂存区。
- .git文件夹除了包括stage暂存区, 还包括git自动为我们创建的第一个分支master, 以及指向master的指针HEAD。

Git读档

- git和读档有关的3个命令
 - git reset --hard HEAD^ (用于读档)
 - git log (用于查看历史版本)
 - git reflog (用于查看未来版本)
- 当我们从Git读档的时候, 我们首先需要让Git知道我们要读哪个档, 在Git中HEAD表示当前版本 (git log出现在最上面的版本), HEAD^表示当前版本的前一个版本, HEAD^^表示当前版本的前两个版本, 如果要回退100个版本, 可以使用HEAD~100。
- 现在git知道了要读哪个档, 那么如何进行读档这个操作呢, 可以使用git reset --hard HEAD^。
- 如果我们回到了HEAD^并关闭了bash, 找不到之前的git log打印出来的commit版本号了, 如何回到最新的档呢?
- git读档的实质是把git内部的HEAD指针指向指定版本。
- 回到当前版本的上一个版本之后, 再使用git log发现最新的版本已经看不到了, HEAD变成了之前的HEAD^, 如果想要再找回之前的最新版本, 我们需要在bash关闭之前去找到上面git log打印出来的最新版本的commit, 再使用git reset --hard切换。
- 我们可以使用git reflog, 这个命令会记录每一次读档操作, 可以从这个命令的输出中找到commit版本号。

查看Git每次修改的日志

- git log命令可以查看git中的存档。
- 为了进行读档, 我们需要知道每一次进行了哪些修改, 以及最近的一次档是什么。
- git就像平时我们在玩RPG游戏的时候, 遇到一关, 我们就会做一个存档, 这样万一操作失误了, 我们可以读取最近的档, 不会让整个游戏从新来过。

创建Git仓库

- 从已有文件夹创建Git仓库
右键选中需要使用Git管理的文件的根文件夹, 选择Git bash here。
在bash输入 git init, git把当前目录变成了一个git仓库, 可以看到当前目录多了一个git文件夹。
- 创建一个空Git仓库
打开git bash, 输入命令mkdir learn git创建一个learn git文件夹。
最后在bash输入git init命令创建一个空git仓库。

把文件添加到git仓库

- 把文件添加到git仓库一共需要2步 (文件需要在git仓库目录下)
 - git add readme.txt (添加一个readme.txt文件到git暂存区)。
 - git commit -m "first commit" (表示把暂存区文件提交到仓库, 字符串是本次提交的说明)。
- 为什么Git添加文件需要add、commit两步呢, 因为commit可以一次提交很多文件, 所以可以多次add不同文件, 在最后输入一个commit命令就会把所有add到缓存区的文件全部提交。
- git commit命令实际上是将暂存区的文件提交到当前分支。

对git仓库中的文件进行修改

- 使用git status查看哪些文件发生了修改
我们可以打开一个已经提交到git仓库的文件进行修改。
- 使用git diff <已修改文件>可以查看对于已修改文件做了哪些修改。(当修改发生在一周之前, 已经记不清做了哪些修改, 可以通过这个命令进行确定再提交到仓库)。
- 对修改后内容提交到仓库还是使用git add添加到暂存区, 再使用git commit -m "second commit"提交到仓库。
- git diff HEAD^ -- readme.txt可以查看上一个档和当前档readme.txt文件的差异。
- 如果git status告诉你存在了文件变动, 那么可以使用git diff查看修改了哪些内容。
- 修改之后在git bash输入git status可以看到当前仓库的状态, 发现readme.txt文件发生了修改但还没有提交。