先推荐一个不错的学习github的博客:
https://blog.csdn.net/qq_35246620/article/details/66973794

本文是基于上面博客里面的内容 进行学习 的;

使用git命令之前先在git里面输入如下命令:

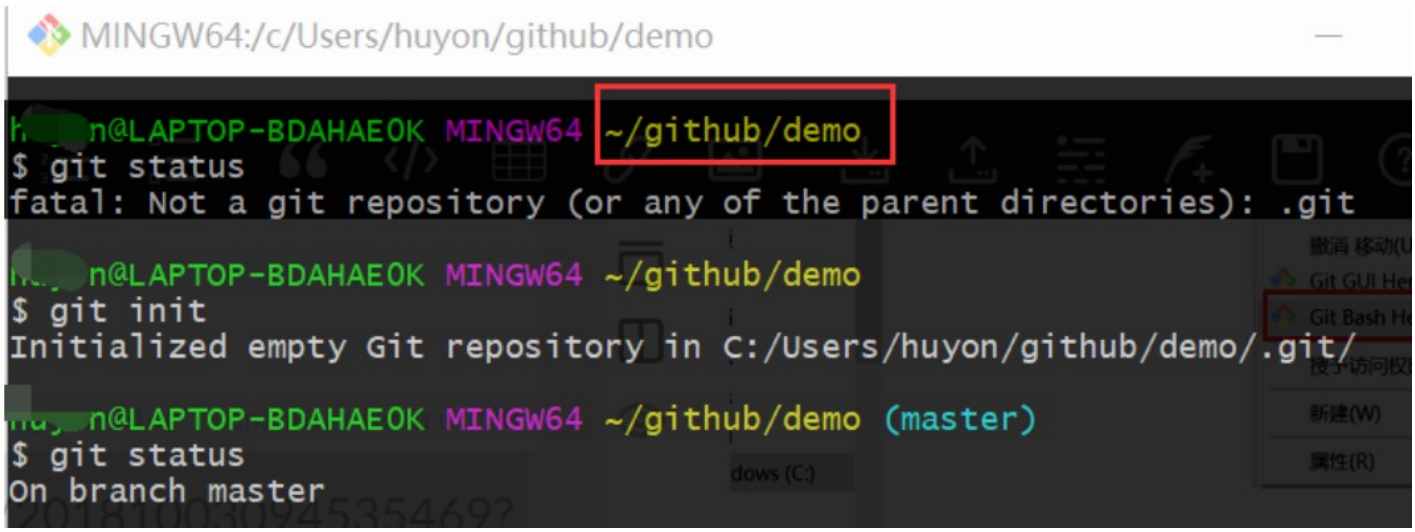
```
请在命令行运行这五句话！！！一定要运行这五句话，不然 git 就不能用了

git config --global user.name 你的英文名字                    #方便产品
git config --global user.email 你的常用邮箱                   #方便产
git config --global push.default simple                       #
git config --global core.quotepath false                      #防
git config --global core.editor "vim"
```

在 Git 中，所有的命令都是以git开头，例如，git init其作用就是初始一个 Git 仓库。

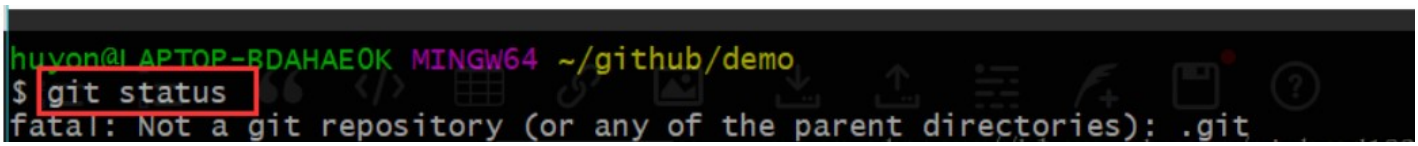
为了方便演示，我们先在C盘的用户名目录下的github目录下创建一个名为demo的子目录,接下来我们的 Git 操作都是基于此目录和文件的;

选择demo目录作为 Git 仓库，然后进入demo目录之中，点击鼠标右键，再选择Git Bash Here，即可打开 Git Bash 的命令行窗口：



第 1 个命令：git status

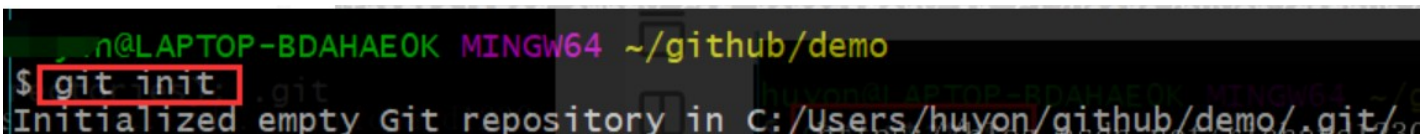
在命令行窗口的光标处，输入git status命令，查看仓库的状态:



如上图所示，结果显示demo不是一个 Git 仓库，这是很正常的反应，因为我们还没有在计算机中声明demo为 Git 仓库;

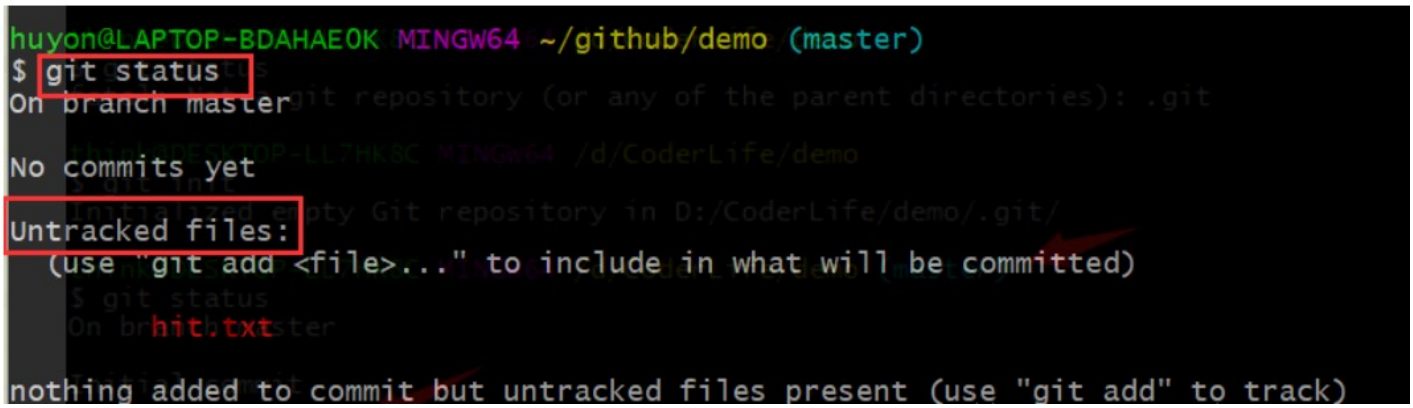
第 2 个命令：git init

在命令行窗口的光标处，输入git init命令，初始化 Git 仓库：



如上图所示，结果显示已经初始化demo为一个空的 Git 仓库啦！

`touch hit.txt` 建立一个文本：



如上图所示，在我们初始化仓库之后，demo目录已经成为一个 Git 仓库了，并且默认进入 Git 仓库的master分支，即主分支。在这里，我们需要注意的是Untracked fies提示，它表示demo仓库中有文件没有被追踪，并提示了具体没有被追踪的文件为hit.txt，还提示了我们可以使用git add命令操作这个文件。

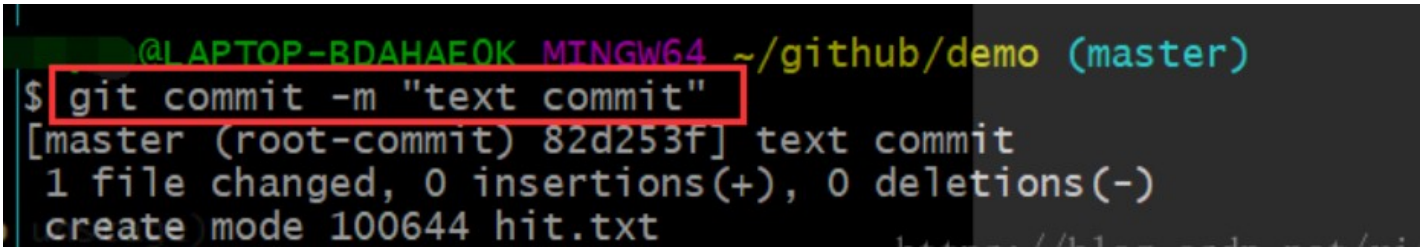
第 3 个命令：git add

在命令行窗口的光标处，输入 `git add hit.txt` 命令，将hit.txt文件添加到 Git 仓库：

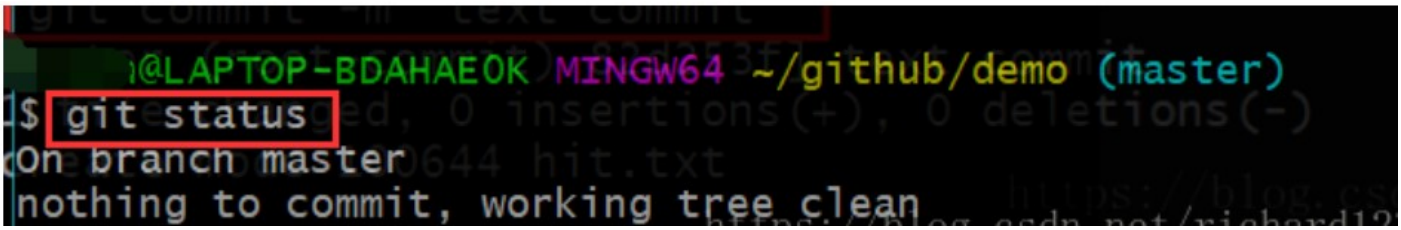
如上图所示，已经显示Initial commit初始化提交了，同时已经没有Untracked files提示了，这说明文件hit.txt已经被添加到 Git 仓库了，而在我们没有进行git add操作之前，文件hit.txt并不被 Git 仓库认可，因此才会出现提示初始化仓库为空的现象。在这里，需要声明一点，那就是：git add命令并没有把文件提交到 Git 仓库，而是把文件添加到了「临时缓冲区」，这个命令有效防止了我们错误提交的可能性。

第 4 个命令：git commit

在命令行窗口的光标处，输入 `git commit -m "text commit"` 命令，将hit.txt文件提交到 Git 仓库：



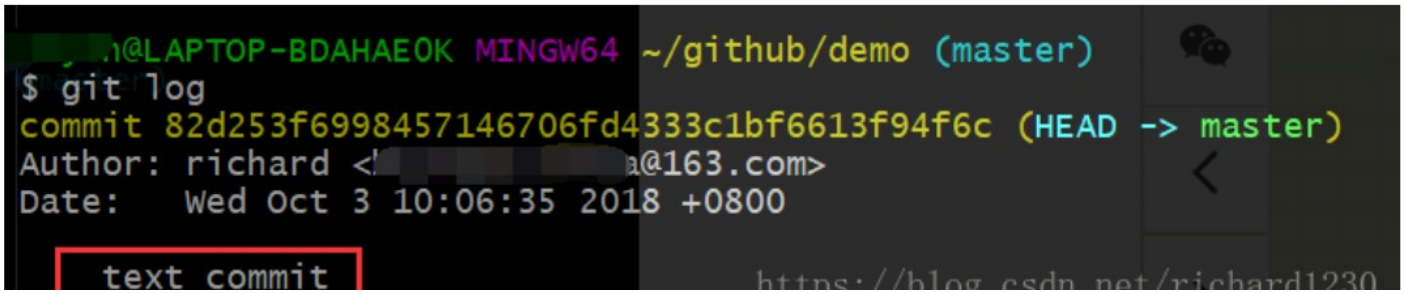
如上图所示，我们成功将文件 `hit.txt` 提交到了 Git 仓库，其中 `commit` 表示提交， `-m` 表示提交信息，提交信息写在双引号""内。接下来，再输入 `git status` 命令查看仓库状态：



如上图所示，结果显示 `nothing to commit, working tree clean` ，这表示已经没有内容可以提交了，即全部内容已经提交完毕。

第5个命令 git log

在命令行窗口的光标处，输入git log"命令，打印 Git 仓库提交日志:



如上图所示，显示了我们的提交记录，提交记录的内容包括Author提交作者、Date提交日期和提交信息。

通过以上的操作，我们会发现一个现象，那就是：在每个git操作之后，我们基本都会输入git status命令，查看仓库状态。这也从侧面说明了git status命令使用的频率之高，也建议大家在操作 Git 仓库的时候多使用git status命令，这能帮助我们实时了解仓库的状态，显然非常有用。

总结

几个常见的命令:
第 1 个命令：git status： 获取当前 仓库状态
第 2 个命令：git init： 初始化仓库
第 3 个命令：git add： 添加文件至仓库,git add命令并没有把文件提交到 Git 仓库，而是把文件添加到了「临时缓冲区」
第 4 个命令：git commit： 提交文件至仓库
第5个命令: git log

第6个命令 git branch

在命令行窗口的光标处，输入git branch命令，查看 Git 仓库的分支情况：
如上图所示，显示了仓库 `demo` 中的分支情况，现在仅有一个master分支，其中 `master` 分支前的 `*` 号表示“当前所在的分支”，例如 `* master` 就意味着我们所在的位置为demo仓库的主分支。输入命令git branch a，再输入命令git branch，结果如下图所示：

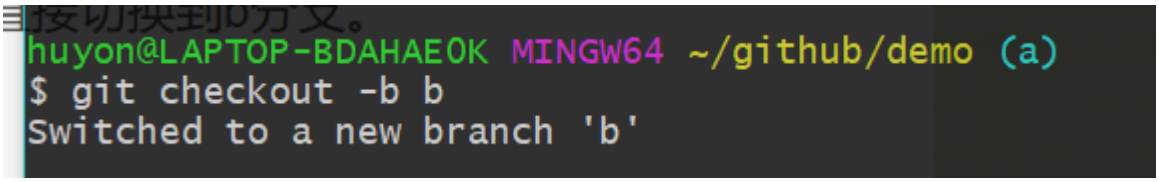
如上图所示，我们创建了一个名为a的分支，并且当前的位置仍然为主分支；

第7个命令：git checkout

在命令行窗口的光标处，输入 `git checkout a` 命令，切换到a分支：

如上图所示，我们已经切换到 `a` 分支啦！也可以通过命令 `git branch` 查看分支情况：

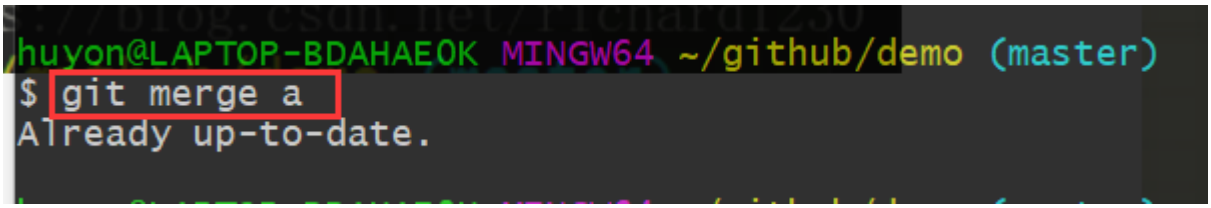
在这里，我们还有一个更简单的方法来查看当前的分支，即通过观察上图中用红色框圈起来的部分。此外，我们也可以在创建分支的同时，直接切换到新分支，命令为 `git checkout -b` ，例如输入 `git checkout -b b` 命令：



如上图所示，我们在a分支下创建b分支（b为a的分支），并直接切换到b分支。

第 8 个命令：git merge

切换到master分支，然后输入git merge a命令，将a分支合并到master分支：



如上图所示，我们已经将a分支合并到主分支啦！此外，在这里需要注意一点，那就是：在合并分支的时候，要考虑到两个分支是否有冲突，如果有冲突，则不能直接合并，需要先解决冲突；反之，则可以直接合并。

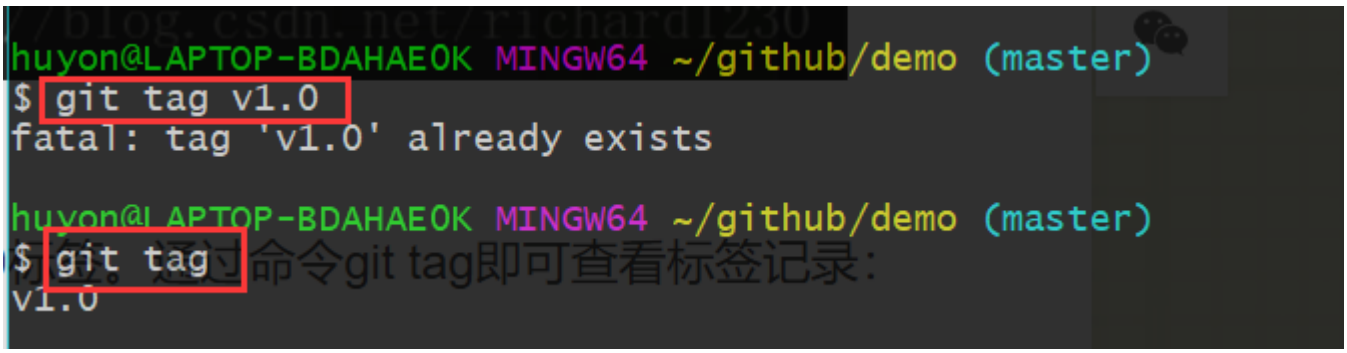
第 9 个命令：git branch -d & git branch -D

在命令行窗口的光标处，输入 `git branch -d a` 命令，删除a分支：

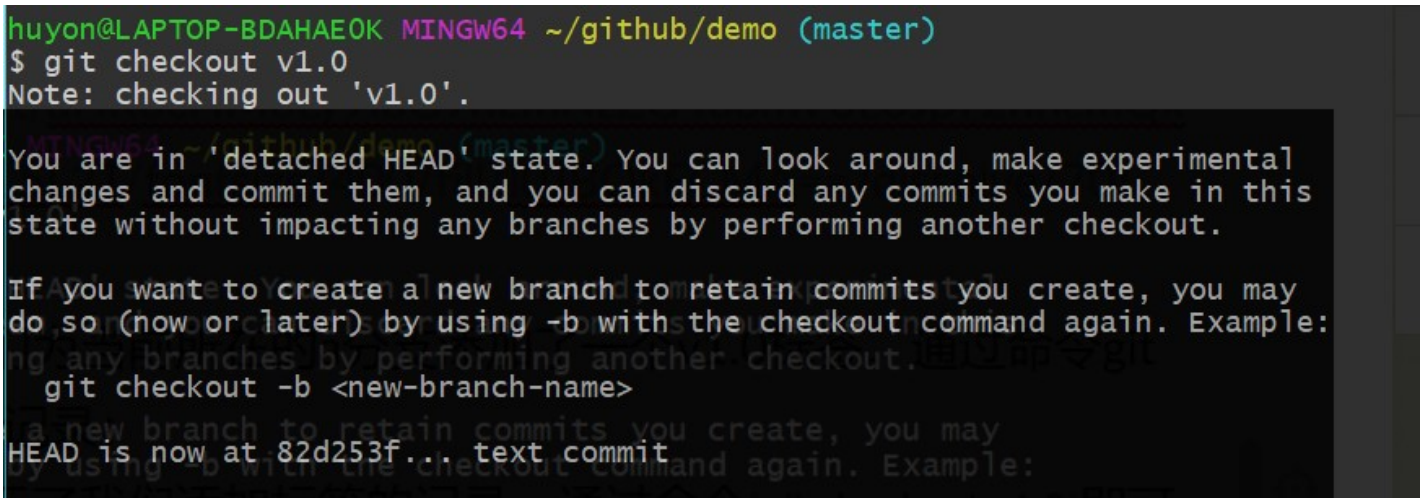
如上图所示，我们已经将分支a删除啦！不过有的时候，通过 `git branch -d` 命令可以出现删除不了现象，例如分支a的代码没有合并到主分支等，这时如果我们一定要删除该分支，那么我们可以通过命令 `git branch -D` 进行强制删除。

第 10 个命令：git tag

在命令行窗口的光标处，输入git tag v1.0命令，为当前分支添加标签：



如上图所示，我们为当前所在的a分支添加了一个v1.0标签。通过命令git tag即可查看标签记录：



如上图所示，显示了我们添加标签的记录。通过命令 `git checkout v1.0` 即可切换到该标签下的代码状态：

通过「Git 初体验及其常用命令介绍」两篇博文的内容，我们已经了解了一些 Git 的常用命令啦，但还有很多命令我们没有进行演示，例如clone、rm、grep、pull和push等，Git 的魅力也并不止于此，还有更多的精彩等待大家探索。