### 2019.07.13学习笔记(a.m)

**Git相关：**

1. **从远程库克隆（即把github上的文件下载到本地）**

Administrator@Helay-computer MINGW64 ~/Desktop

$ git clone git@github.com:Helayxy/gitskills.git

Cloning into 'gitskills'...

remote: Enumerating objects: 3, done.

remote: Counting objects: 100% (3/3), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Receiving objects: 100% (3/3), done.

1. 要克隆一个仓库，首先必须知道仓库的地址（Helayxy/gitskills.git）；
2. 使用git clone进行克隆；
3. Git支持多种协议，包括https，但通过ssh支持的原生git协议速度最快。
4. **分支相关命令：（<name>表示分支名）**

1）查看分支：$ git branch

2）创建分支：$ git branch <name>

3）切换分支：$ git checkout <name>

4）创建并切换分支：$ git checkout -b <name>（相当于同时执行2）3）两个命令）

5）创建+切换分支：$ git checkout -b <name>

6）合并某分支到当前分支：$ git merge <name>

7）删除分支：$ git branch -d <name>

1. **分支相关操作：**

**1）创建dev分支，然后切换到dev分支:**

$ git checkout -b dev

Switched to a new branch 'dev'

**2）用git branch命令查看当前分支：**

$ git branch

\* dev

Master

**注：git branch命令会列出所有分支，当前分支前面会标一个\*号。**

**在当前分支对文件进行修改和在master分支修改效果一样，此时的dev分支就是master分支的一个克隆分支，两者一模一样，只是在dev分支修改问件事，master分支不受影响，还保持原来的状态**

Administrator@Helay-computer MINGW64 ~/Desktop/gitskills (dev)

$ vi test.txt

（vi操作为在test.txt文件上新添加了一行wwww，原文件只有一行qqqq）

**3）添加、提交：**

$ git add test.txt

Administrator@Helay-computer MINGW64 ~/Desktop/gitskills (dev)

$ git commit -m "add a line"

[dev 76bbcb9] add a line

1 file changed, 1 insertion(+)

**4）dev分支的工作完成，我们就可以切换回master分支：**

$ git checkout master

Switched to branch 'master'

Your branch is up to date with 'origin/master'.

**5）查看test.txt内容：**

$ cat test.txt

qqqq

**注：发现还是只有qqqq，并没有新添加的一行wwww**

**（原因：只是在dev分支对文件进行了修改，master分支状态不变）**

**6）把dev分支的工作成果合并到master分支上：**

$ git merge dev

Updating 7a73e75..76bbcb9

Fast-forward

test.txt | 1 +

1 file changed, 1 insertion(+)

**注：Fast-forward信息，Git告诉我们，这次合并是“快进模式”，也就是直接把master指向dev的当前提交，所以合并速度非常快（也不是每次合并都能Fast-forward）。**

**7）再次查看test.txt内容：**

$ cat test.txt

qqqq

wwww

（新添加的wwww一行已经可以在master分支上显示了）

**8）合并完成后，删除dev分支：**

$ git branch -d dev

Deleted branch dev (was 76bbcb9).

**9）查看分支：**

$ git branch

\* master

**只剩下master分支了**

1. **分支冲突相关操作：**

**当在新建的分支上做了修改并提交了，再切换到master分支，这时在master分支上再对文件进行修改，最后使用git merge命令合并时会产生冲突，如下所示：**

$ git merge feature1

Auto-merging test.txt

CONFLICT (content): Merge conflict in test.txt

Automatic merge failed; fix conflicts and then commit the result.

**显示合并失败！**

**这时就需要手动去解决冲突。**

**1）当合并产生冲突时，查看文件，会看到不同分支指向的内容：**

Administrator@Helay-computer MINGW64 ~/Desktop/gitskills (master|MERGING)

$ cat test.txt

qqqq

<<<<<<< HEAD

Creating a new branch is quick & simple.

=======

Creating a new branch is quick AND aimple.

>>>>>>> feature1

**注：Git用<<<<<<<，=======，>>>>>>>标记出不同分支的内容**

**2）使用$ git log --graph --pretty=oneline --abbrev-commit可以查看分支图：**

$ git log --graph --pretty=oneline --abbrev-commit

\* 0f8061e (HEAD -> master) conflict fixed

|\

| \* 6bac58d (feature1) AND simple

\* | e194fca & simple

|/

\* 76bbcb9 add a line

\* 7a73e75 (origin/master, origin/HEAD) add test.txt

\* 02374b6 Initial commit