自己的github <https://github.com/lpt999/firstDemo>

用户名 ： lpt999

密码 ： li9432913

设置全局用户名 ：git config --global user.name ‘\*\*\*\*’;

设置邮箱 ： git config –global user.email ‘\*\*\*@126.com’;

## 一、git管理本地项目：

1. 创建一个本地项目 🡪 git init --bare;
2. 克隆一个项目到本地 🡪 git clone 远程项目路径 . (.代表当前路径);
3. 将一个新添加的文件纳入版本管理之中 🡪 git add 文件名;
4. 将文件提交到本地版本管理之中 🡪 git commit 文件名;
5. 将本地文件提交到远程项目 🡪 git push 远程项目路径 master;
6. 刷新远程代码到本地 🡪 git pull;
7. 查看本地的项目的状态 🡪 git status;
8. 查看本地修改 🡪 git diff 文件名；
9. 查看下历史记录 🡪 git log；
10. 版本回退操作：git reset  –hard HEAD^ ，如果要回退到上上个版本只需把HEAD^ 改成 HEAD^^ 以此类推。那如果要回退到前100个版本的话，使用上面的方法肯定不方便，我们可以使用下面的简便命令操作：git reset  –hard HEAD~100 即可
11. 我们可以通过版本号回退，使用命令方法如下：git reset  –hard 版本号 ；
12. 丢弃工作区的修改 git checkout  -- file；
13. 删掉文件 ： rm file；然后 git commit；

## 二、git管理远程仓库：

1、在本地的testgit仓库下运行命令：

git remote add origin 远程链接地址；

git push origin master

2、从远程库克隆

git clone远程链接地址 .；

3、git checkout 命令加上 –b参数表示创建并切换，相当于如下2条命令

git branch dev

git checkout dev

 查看分支：git branch

   创建分支：git branch name

   切换分支：git checkout name

创建+切换分支：git checkout –b name

合并某分支到当前分支：git merge name

删除分支：git branch –d name

4、分支管理策略。

      通常合并分支时，git一般使用”Fast forward”模式，在这种模式下，删除分支后，会丢掉分支信息，现在我们来使用带参数 –no-ff来禁用”Fast forward”模式。首先我们来做demo演示下：

1. 创建一个dev分支。
2. 修改readme.txt内容。
3. 添加到暂存区。
4. 切换回主分支(master)。
5. 合并dev分支，使用命令 git merge –no-ff  -m “注释” dev
6. 查看历史记录

5、

bug分支：

     在开发中，会经常碰到bug问题，那么有了bug就需要修复，在Git中，分支是很强大的，每个bug都可以通过一个临时分支来修复，修复完成后，合并分支，然后将临时的分支删除掉。

比如我在开发中接到一个404 bug时候，我们可以创建一个404分支来修复它，但是，当前的dev分支上的工作还没有提交。比如如下：



   并不是我不想提交，而是工作进行到一半时候，我们还无法提交，比如我这个分支bug要2天完成，但是我issue-404 bug需要5个小时内完成。怎么办呢？还好，Git还提供了一个stash功能，可以把当前工作现场 ”隐藏起来”，等以后恢复现场后继续工作。如下：



   所以现在我可以通过创建issue-404分支来修复bug了。

首先我们要确定在那个分支上修复bug，比如我现在是在主分支master上来修复的，现在我要在master分支上创建一个临时分支，演示如下：



修复完成后，切换到master分支上，并完成合并，最后删除issue-404分支。演示如下：



现在，我们回到dev分支上干活了。



工作区是干净的，那么我们工作现场去哪里呢？我们可以使用命令 git stash list来查看下。如下：



工作现场还在，Git把stash内容存在某个地方了，但是需要恢复一下，可以使用如下2个方法：

1. git stash apply恢复，恢复后，stash内容并不删除，你需要使用命令git stash drop来删除。
2. 另一种方式是使用git stash pop,恢复的同时把stash内容也删除了。

         演示如下



八：多人协作。

当你从远程库克隆时候，实际上Git自动把本地的master分支和远程的master分支对应起来了，并且远程库的默认名称是origin。

1. 要查看远程库的信息 使用 git remote
2. 要查看远程库的详细信息 使用 git remote –v

如下演示：

