

## 单人流程:

### 一、准备工作(只做一次):

1. 创建一个工作区
2. 在工作区中的打开git终端
3. 通过 `git init` 指令, 初始化版本库
4. 设置用户名和邮箱
  - `git config user.name "姓名"`
  - `git config user.email "邮箱"`
5. 通过 `git config -l` 查看设置情况

### 二、开发阶段(反复执行)

1. 编写代码
2. 通过 `git add 文件名称(可用通配符, 例如*)` 或者 `git add .` 添加到版本库的暂缓区中
3. 通过 `git commit -m "说明"` 将暂缓区的文件添加到HEAD指针指向的分支中(默认只有一个分支, master分支, 也称之为主分支)

#### 注意点:

1. 不是写一句代码就 `add commit` 一次, 应该是完成一个功能后再 `add commit`
2. `commit` 时 `-m` 注释一定要认真编写, 与当前提交内容保持一致

### 单人使用Git管理项目好处:

1. 可以通过 `git status` 查看哪些文件没有被管理, 修改了哪些文件红色(没有被管理或者被修改了)、绿色(在暂缓区)
2. 可以通过 `git diff` 查看具体修改了哪些代码
3. 可以通过 `git log / git reflog` 查看项目演变历史
4. 可以通过 `git reset--hard 版本号` 在任意版本之间切换
5. 无需备份多个文件, 每次 `commit` 提交Git会自动备份

## 多人开发:

## 一、在远程服务器上创建一个共享版本库

1. 项目负责人打开远程的服务器，然后创建一个工作区
2. 在远程的服务器上打开工作区，在工作区中打开Git终端工具
3. 在Git终端工具中输入 `git init --bare`
4. 经过以上几步，就代表远程服务器上的共享版本库已经创建好了

## 二、开发人员下载远程版本库

1. 开发人员在自己的电脑上打开Git终端工具
  2. 从远程的服务器上下载当前项目的共享版本库 `git clone 远程服务器共享版本库地址`。
- 和单人开发使用Git的区别: 单人开发是自己创建版本库，而多人开发是从远程服务器下载版本库

## 三、进入开发阶段

和单人开发一样

1. 设置用户名和邮箱
2. 编写代码
3. `git add`. 添加到暂缓区
4. `git commit -m` 添加到 HEADER 指针指向的分支
5. 将代码提交到远程的服务器 `git push`
6. 其它的开发人员通过 `git pull` 拿到更新的代码

## 多人开发使用Git注意点:

1. 不能将不能运行的代码提交到本地和远程服务器(切记一定不能)
2. 如果服务器上有其它开发人员的更新内容，则不能直接将我们的代码 `push` 到服务器。要先将其它开发人员更新的内容 `pull` 到本地之后才能 `push`。
3. 如果我们更新的内容和其它同事更新的内容有冲突(修改了同一个文件的同一行代码)，需要手动解决冲突。

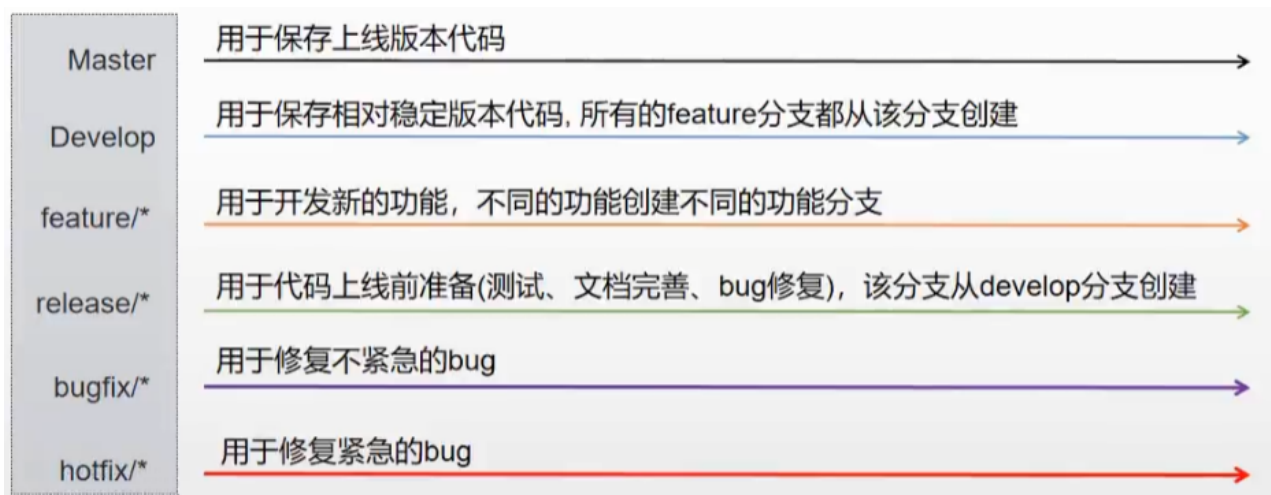
开发技巧:

只要开发完了一个功能就要立即提交代码，因为在企业开发中谁后提交谁就负责解决冲突。

## 分支

- `git branch`  
查看所有分支
- `git branch -r`  
查看远程仓库有哪些分支

- **git branch 分支名称**  
创建分支（新创建的分支中的内容和master分支中的内容一样）
- **git check out 分支名称**  
切换到指定分支
- **git switch 分支名称**  
切换到指定分支
- **git merge 分支名称**  
合并分支（将当前所在分支和指定名称分支进行合并）
- **git branch -d 分支名称**  
删除指定分支（不能在当前分支中删除自己）



## GitCheetSheet



## 终端常用指令

命令	说明	参数缺省时
mkdir [DIR_NAME]	创建目录	
cd [DIR_NAME]	进入目录	
ls [DIR_NAME]	查看目录内容	查看当前目录内容
cat [FILE_NAME]	查看文件内容	

## 新建项目

[https://blog.csdn.net/m0\\_52711790/article/details/119566784](https://blog.csdn.net/m0_52711790/article/details/119566784)

- 1 git init // 1.将项目文件夹初始化为git仓库
- 2 git add . // 2.把文件添加到暂存区
- 3 git commit -m 'first commit' // 3.把文件提交到仓库，引号内为说明内容
- 4 git remote add origin 你的远程库地址 // 4.关联到远程库
- 5 git pull --rebase origin master // 5.获取远程库与本地同步合并
- 6 git push -u origin master // 6.把本地库的内容推送到远程

## 一般流程

1.进入项目根目录

```
$ cd d:/project/demo
```

2.初始化仓库

```
$ git init
```

3.然后添加到远程仓库

(1) 在git里面创建一个仓库

(2) 获取该仓库的SSH地址

(3) 命令行中输入添加到远程仓库的命令

```
git remote add origin git@gitee.com:kitecat/c-language.git
```

4.再将代码添加到本地git

```
$ git add .
```

5.将代码提交到本地代码库

```
$ git commit -m "提交命名, 可任意"
```

6.最后将本地git推到远程上

```
$ git push origin master
```

```
$ git push -u origin master (第一次用)
```

7.如果失败(error: failed to push some refs to 'github.com:Folo92/xxx.git'), 先把远程库同步到本地

```
$ git pull origin master
```

8.如果失败(fatal: refusing to merge unrelated histories)

```
$ git merge origin master --allow-unrelated-histories
```

9.然后再提交

```
$ git commot -m 'xxx'
```

10.推送

```
$ git push -u origin master
```

## 修改项目名称

<https://letmefly.blog.csdn.net/article/details/124348891>

```
1 //先修改原来的origin为old-origin
2 git remote rename origin old-origin
3 //添加新的origin
4 git remote add origin git@github.com:LetMeFly666/MatFeature.git
```

```
5 //将新地址设为默认
6 git pull -u origin master
```

- 1.可以直接在Github上修改，并且本地不做任何更改。这样不好。
2. git remote rename origin old-origin 重命名旧地址
3. git remote add origin git@github.com:LetMeFly666/MatFeature.git 添加新地址
4. git pull -u origin master 将新地址设为默认。

## 版本回退

### git reset 的三种模式

命令	说明	工作区内容	暂存区内容
git reset --soft	软回退	保留	保留
git reset --hard	硬回退	回退	回退
git reset --mixed	混合回退（默认）	保留	回退

## 问题

1、Git执行命令git config --list 后卡住无法操作，一直显示end，按回车也没有用

解决方法：按<Enter> + <Q> || <Enter>

2、fatal: unable to access 'https://github.com/...' : OpenSSL SSL\_read: Connection was reset, errno 10054

产生原因：一般是这是因为服务器的SSL证书没有经过第三方机构的签署，所以才报错

参考网上解决办法：解除ssl验证后，再次git即可

git config --global http.sslVerify false

3、查看 more id\_rsa.pub 密钥

在.ssh文件夹打开cmd，输入 more id\_rsa.pub 回车

4、如何解决 Git 中的合并冲突——示例的实用指南

[https://zhuanlan.zhihu.com/p/515478525?utm\\_id=0](https://zhuanlan.zhihu.com/p/515478525?utm_id=0)

5、git log 和 git reflog 的区别

[https://blog.csdn.net/Seky\\_fei/article/details/114729213](https://blog.csdn.net/Seky_fei/article/details/114729213)

## 创建仓库

git init	初始化仓库
git clone	拷贝一份远程仓库，也就是下载一个项目。

## 提交与修改

git add .	添加文件到暂存区
git status	查看仓库当前的状态，显示有变更的文件。
git diff	比较文件的不同，即暂存区和工作区的差异。
git commit	提交暂存区到本地仓库。
git reset	回退版本。
git rm	将文件从暂存区和工作区中删除。
git mv	移动或重命名工作区文件。
git branch -m <old> <new>	修改本地分支名称

## 提交日志

git log	查看历史提交记录
git blame <file>	以列表形式查看指定文件的历史修改记录

## 远程操作

git remote	远程仓库操作
git remote add <name> <url>	连接远程仓库
git fetch	从远程获取代码库
git pull	下载远程代码并合并
git push	上传远程代码并合并

## 其他操作

--	--

快捷键

Ctrl+Ins 复制

Shift+Ins 粘贴