

Commit规范

1. [Fix bugs] 修改了哪些存在的问题
2. [New features] 新增了哪些功能
3. [Enhancement] 完善了哪些功能
4. [Merge] 合并了什么功能

常用命令

创建仓库

1. 初始化仓库: `git init`
2. 克隆仓库: `git clone 仓库地址`

远程仓库相关

一个本地仓库可以关联多个远程仓库，默认会有一个origin的远程仓库。当本地仓库需要与多个远程仓库关联时（比如一个本地仓库需要push到多个远程仓库上），可以用 `git remote -add` 添加新的仓库地址并且指定他的仓库名，之后push和pull的操作就可以针对指定的仓库进行。

1. 添加远程仓库: `git remote add 仓库名 仓库地址`
2. 查看所有关联的远程仓库: `git remote`
3. 查看所有关联的远程仓库（包括地址）: `git remote -v`

仓库状态相关

1. 查看仓库状态: `git status`
2. 查看commit记录: `git log`

Add操作相关

1. add所有更改：git add .
2. 撤销当前add操作：git reset HEAD
3. 撤销对某个文件的当前add操作：git reset HEAD 文件路径

注：涉及到撤销操作需**慎重**，最好多百度一下

1. 添加更改到暂存区：git stash（一般希望丢弃掉一些不必要的更改时就用这个命令。比如你稍微修改了一点内容，这些内容是不必要的，比如加了几句测试代码，同时这时候你需要pull别人的分支，这时候就可以使用git stash来“丢弃”你自己的代码，这样会比较方便。但这个命令有它的具体用途，见：[git-stash用法小结 - Tocy - 博客园 \(cnblogs.com\)](https://cnblogs.com))

Commit操作相关

1. commit所有更改：git commit -m “备注内容”
2. 修改上次commit备注内容：git commit --amend（这里是使用vim的操作方式，按i修改内容，修改完后按Esc返回，然后输入“:wq”+回车保存修改）

注：git很多地方使用的是vim的编辑模式，如果发现在某个页面无论怎么操作都没反应，可以试试输入“:q”+回车就可以退出

1. 撤销当前commit操作（但不改变文件内容）：git reset --soft HEAD^
2. 撤销当前commit操作（改变文件内容）：git reset --hard HEAD^

注：涉及到撤销commit操作需要**更加慎重**！最好多百度一下，可以参考：[git commit后，如何撤销commit - 简书 \(jianshu.com\)](https://jianshu.com)

1. 回退到指定的历史commit：git reset --hard CommitID

注：具体见：[git reset --hard HEAD - 简书 \(jianshu.com\)](https://jianshu.com)

分支操作相关

1. 添加并切换到新分支：git checkout -b 分支名
2. 切换分支：git checkout 分支名
3. 查看分支（本地分支）：git branch
4. 查看所有分支（包括远程仓库上的分支）：git branch -a

remotes/仓库名/分支名 就是远程分支

合并操作相关

1. 合并分支: `git merge 分支名`
2. 中断分支合并: `git merge --abort`

注意事项: [git merge --abort的一种使用情况_磊怀的博客-CSDN博客](#)

1. 解决分支合并冲突: `git mergetool`

注: 这一步通常需要安装一个可视化的合并工具, 可以安装p4vmerge, 自己也可以找一下其他的工具

Pull/Push操作相关

1. 拉取代码: `git pull 仓库名 (默认为origin) 分支名`

注1: 还有一个命令fetch, 不过我用的不多, 我一般直接pull, 如果当前分支与pull下来的分支不同, 就pull下来之后再merge, 否则就会自动merge

注2: 如果pull报错, 一般就是因为本地有还没有保存的更改, 通过先把当前的更改commit了。如果这些更改可以被丢弃, 就执行git stash

1. 提交代码: `git push 仓库名 (默认为origin) 分支名`

注: 如果push报错, 除了网络原因外, 一般就是因为远程仓库与你的本地仓库的当前“版本”不一致, 需要先把代码pull下来合并后再push

其他

1. 导出代码压缩包: `git archive -o "文件路径" 分支名`

如: `git archive -o "source.zip" master`: 把master分支的代码压缩到source.zip中

一般流程

1. 下载仓库: `git clone 仓库地址`

具体说明

1. 在你要创建项目的地方，右键打开git bash，运行

```
1 | git clone ``https://gitee.com``/.../xxx``.git
```

即可把仓库下载到本地，具体要clone的仓库链接自行在gitee上面复制

1. 执行 `cd xxx`，进入工程目录（XXX表示git在你本地创建出来的工程文件夹名）
2. 执行 `git checkout -b 分支名` 新建并切换到自己的分支，分支名用自己名字首字母即可，比如：`git checkout -b lja`
3. 上传更改：

`git add .`

`git commit -m "备注内容"`

`git pull 仓库名 分支名`

`git push 仓库名 分支名`

具体说明

1. 确保当前是你自己的分支
2. 执行 `git add .`
3. 执行 `git commit -m "备注内容"` 提交你的更改
4. （该步骤可选）执行 `git pull origin master`，拉取并合并master分支的修改（PS. 合并时有可能会遇到冲突。遇到冲突时，如果自己没信心解决的，就不要执行该步骤了，使用 `git merge --abort` 终止操作）
5. 执行 `git push origin 你自己的分支名`，就能提交了
6. 在修改了一些代码后（还没有commit）才记得要在一个新分支上开发：

`git checkout -b 分支名`

`git add .` // 可选，如果你不想commit就不管

`git commit -m "备注内容"`

注意，checkout命令中，只有创建新分支才需要“-b”，如果是已有的分支直接checkout即可

1. 在修改了一些代码后（已经commit了）才记得要在一个新分支上开发：

```
git reset --soft HEAD^
```

```
git checkout -b 分支名
```

```
git add . // 可选，如果你不想commit就不管
```

```
git commit -m "备注内容"
```

1. 在修改了一些代码后（已经commit了），需要push到一个新分支上：

```
git checkout -b 新分支名
```

```
git merge 原分支名
```

```
git push 仓库名 新分支名
```

1. 当需要多人协作时，除了gitee外还需要用到新仓库时候，将当前代码push到新仓库上：

```
git remote add 仓库名 新仓库地址
```

```
git push 仓库名 分支名
```

以github仓库，master分支为例：

```
git remote add github https://github.com/....
```

```
git push github 分支名
```

1. 当你本地改了一些内容，但是这些内容是可以丢弃的，因为别人已经有了更完善的版本，这时候你要把别人的代码拉下来：

```
git stash
```

```
git pull 仓库名 分支名
```

1. 误添加了不需要同步的某些大文件，而且已经commit过很多次了，从已经commit了的历史记录中删除这些大文件：[如何删除git提交历史中包含的大文件? - 知乎\(zhihu.com\)](#)，或者使用bfg（更加推荐）：[BFG Repo-Cleaner - 快速清除Git提交历史中的特定文件 - 云+社区 - 腾讯云\(tencent.com\)](#)，英文官网：[BFG Repo-Cleaner by rtyley](#)，但一般不允许擅自进行此操作

注：记得先配置好.gitignore文件