1、本地查看853769620

git diff：工作区和暂存区

git diff --cached：暂存区和仓库

git diff HEAD：工作区和仓库

git diff feature/001 master：两个分支之间的差异

git diff HEAD HEAD^：最近两次提交之间的差异

我在工作中常用的两个技巧，什么时候要去对比不同版本的代码差异呢？上一讲其实重点是关注的比较粗的层面，差了多少个commit

比如说我今天开发了一些代码，还没提交，或者是已经提交了，此时我希望看一下，确认一下今天跟昨天的代码之间的差异是什么？

就是我告诉大家，大家作为一个程序员，码农，我喜欢的是那种心细如发的码农，很细心的人

不细心的人，很容易造成各种乱七八糟的bug

今天开发了一些代码，回头看看，跟昨天的比较一下，看看我到底加了哪些代码，我新增的这些代码会不会有什么bug？

我的一个工作习惯，每天必看今天做出的代码差异，确保这些代码差异没什么大问题，仔细反复脑子了复盘几次，梳理几遍，想想这么写代码有没有问题，有没有其他更好的方式去写代码

其实这个过程，相当于是你对今天的工作做一个复盘，相当于是你对自己今天的代码做一个code  review

随便举几个例子，如何对自己的代码进行查漏补缺：

（1）看看，注释有没有写全？每个方法，每个类，类里面的核心代码，注释有没有写全

（2）类命名，方法命名，有没有什么问题

（3）有没有什么方法，圈复杂度过高，圈复杂度，如果你的方法里面，if else里面嵌套了一个if else，再次嵌套if else，嵌套了十来层，圈复杂度过高，重构一下方法，有些if else需要抽取出来做成单独的方法，便于维护

（4）你写的代码扩展性怎么样，如果以后要扩展的话，好不好扩展，能不能用什么设计模式来重新设计一下？

今天和昨天写的代码的差异看一下

（1）今天写的代码还没有做commit，此时的话，代码还停留在工作区中，此时用一个命令：git diff HEAD，将工作区中的代码和仓库中最近一次commit的代码做比较，给你显示出来差异

（2）今天写的代码已经做额commit，此时可以用：git diff HEAD^ HEAD，比较commit到仓库中的代码，最近两个commit之间的代码差异

2、gitlab上查看两个分支之间的差异

还有别的场景

（1）你现在在写一个feature分支，你想看一下feature分支和master分支之前的差异，git diff  master feature/001

（2）这个过程同样是相当于你自己在做code review，就是说不一定每天都干，但是，最好是每隔一段时间，都仔细看一下你最近做的代码修改和master之间的差异，确保说你的代码没有很大的问题，对你写的代码仔细的去思考，有没有什么大问题