* git reset命令选项

① git reset --mixed HEAD~：默认，移动HEAD指向到上一个版本，并将HEAD移动后指向的快照回滚到暂存区

② git reset --soft HEAD~：只移动HEAD指向到上一个版本

③ git reset --hard HEAD~：移动HEAD指向到上一个版本，将HEAD移动后指向的快照回滚到暂存区，而且将暂存区的文件还原到工作区

④ git reset commit\_id：回滚到指定id(SHA1值)的快照

* GitHub Pages搭建网站

访问：https://用户名.github.io

搭建步骤：

①创建个人站点：新建仓库(仓库名必须是[用户名.github.io])

②在仓库下新建index.html文件作为首页

注意：GitHub Pages仅支持静态网页；仓库里只能是html文件

Project Pages项目站点

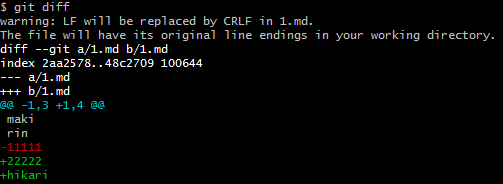
访问：https://用户名.github.io/仓库名

项目主页Settings--> GitHub Pages-->选择Source和Theme，各种确认就OK



* 版本对比

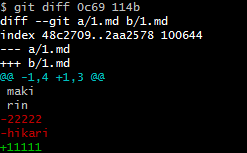
①git diff：不带选项为对比工作区和暂存区



---表示在暂存区的旧文件；+++表示工作区新修改的文件

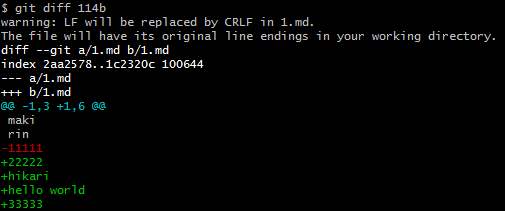
② git diff commit\_id1 commit\_id2：比较两个历史快照

使用git log查看不同快照的版本号，再比较



③ git diff commit\_id：比较当前工作区与Git仓库历史某个id的快照

比如git diff 114b：比较工作区与第1次提交的版本



比较工作区与最近一次提交版本：git diff HEAD

④ git diff --cached [commit\_id]：比较暂存区和Git仓库快照

不写id表示暂存区和最近一次提交版本对比；也可以指定版本号id

* 修改最后一次提交

版本刚提交(commit)到仓库，①发现有几个文件没有add；②版本说明写得不够全面，无法彰显此次修改的意义…

git commit --amend：更正最近一次提交

比如第3次修改后提交描述直接写333很不直观，需要修改：

输入命令后跳到修改界面(vim操作)，修改完：



再次查看git log，还是只有3条记录，修改了最后一条的说明，没有新增快照

* 重命名文件：git mv old\_name new\_name

相当于①ren/mv old\_name new\_name②git rm old\_name③git add new\_name