## git 使用说明

git操作笔记，初级使用教程请参考 [git cheatsheet](#git-cheatsheet) 。

### 添加文件

git add .

### 删除文件

git rm .

### 提交改动到本地

git commit -m "first commit"

### 上传改动到服务器

git push

### 列出当前分支

git branch

### 列出所有分支，包括远程分支

git branch -a

### 从已有的分支创建新的分支(如从master分支),创建一个dev分支

git checkout -b dev

### 从服务器拉取分支到本地分支

git pull origin gh-pages:gh-pages

需要本地没有gh-pages分支，否则会提示已拒绝 ### 上传本地分支到远程分支

git branch --set-upstream-to=gh-pages  
git push origin gh-pages

## gerrit

如何修改一个commit：

1. 先在一个干净的repo里面把这一个commit 拉取下来：

git fetch ssh://xxx@xxxx/xxx/xxx refs/changes/20/190820/6 && cherry-pick FETCH\_HEAD

具体指令需要看gerrit网页右上角：



此时git status 当前是有y一个commit的，git log也能看到log。

1. 修改代码，然后提交到本地。

git add  
#使用 --amend 重新更新这个commit信息。  
git commit --amend

1. push到gerrit

需要指定具体push到哪个id

git push origin HEAD:refs/changes/<change id>

以下记录一些底层 git 技术。

### clone 本地仓库

$ git clone /home/git/repositories/801/038/000/38801.git

### 查看object文件的内容

使用 git cat-file 可以查看 object 文件的内容，但是查看的内容是转换过的：

$ git cat-file -p aa548c4d7910229712ba3a41e74c6db872e8ab64  
100644 blob c30106543ed8f32af334362fa82e3a4ad71ef20f home.md

可以使用 [zlib-flate](http://manpages.ubuntu.com/manpages/trusty/man1/zlib-flate.1.html) 命令解压看到真实内容：

$ zlib-flate -uncompress < .git/objects/aa/548c4d7910229712ba3a41e74c6db872e8ab64 | hexdump -C  
00000000 74 72 65 65 20 33 35 00 31 30 30 36 34 34 20 68 |tree 35.100644 h|  
00000010 6f 6d 65 2e 6d 64 00 c3 01 06 54 3e d8 f3 2a f3 |ome.md....T>..\*.|  
00000020 34 36 2f a8 2e 3a 4a d7 1e f2 0f |46/..:J....|  
0000002b

tree 对象 文件格式请看：[tree 对象](https://juejin.im/post/6874840619332665357#heading-16)。 需要安装 qpdf 才能正常使用 [zlib-flate](http://manpages.ubuntu.com/manpages/trusty/man1/zlib-flate.1.html) 命令：apt install qpdf 。

### git cat-file

查看objects文件。 **说明**

usage: git cat-file (-t [--allow-unknown-type] | -s [--allow-unknown-type] | -e | -p | <type> | --textconv | --filters) [--path=<path>] <object>  
 or: git cat-file (--batch | --batch-check) [--follow-symlinks] [--textconv | --filters]  
  
<type> can be one of: blob, tree, commit, tag  
 -t show object type  
 -s show object size  
 -e exit with zero when there's no error  
 -p pretty-print object's content  
 --textconv for blob objects, run textconv on object's content  
 --filters for blob objects, run filters on object's content  
 --path <blob> use a specific path for --textconv/--filters  
 --allow-unknown-type allow -s and -t to work with broken/corrupt objects  
 --buffer buffer --batch output  
 --batch[=<format>] show info and content of objects fed from the standard input  
 --batch-check[=<format>]  
 show info about objects fed from the standard input  
 --follow-symlinks follow in-tree symlinks (used with --batch or --batch-check)  
 --batch-all-objects show all objects with --batch or --batch-check

**示例**

# 查看objects文件类型  
$ git cat-file -t 56ec1a0729533fbd8d38b7964b6f8ca2cace70ba  
commit  
  
# 查看objects文件大小  
$ git cat-file -s 56ec1a0729533fbd8d38b7964b6f8ca2cace70ba  
243  
  
# 查看objects文件（格式化）内容  
$ git cat-file -p 56ec1a0729533fbd8d38b7964b6f8ca2cace70ba  
tree 7f9adb36c3e987d1ca9d40ba538afd8cbc74e942  
parent cf22ff3d15d718603f93c36f13d848e00d841def  
author chenan.xxw <chenan.xxw@alibaba-inc.com> 1599545250 +0800  
committer chenan.xxw <chenan.xxw@alibaba-inc.com> 1599545250 +0800  
  
update README.md  
  
# 也可以通过 tag 或 branch 名称查看 object 文件  
$ git cat-file -t v0.1.0  
tag  
$ git cat-file -p v0.1.0  
object c5b97d5ae6c19d5c5df71a34c7fbeeda2479ccbc  
type commit  
tag v0.1.0  
tagger Scott Chacon <schacon@gmail.com> 1290556430 -0800  
  
tagging initial release of libgit2

> [Gitlab](http://gitlab.com/) 官方提供的 git cheatsheet 。