# git log

目录


- 命令行参数_____
  - --oneline


`git log` 命令按照提交时间从最晚到最早的顺序，列出所有 commit。

```
# 列出当前分支的版本历史
$ git log


# 列出某个文件的版本历史，包括文件改名
$ git log --follow [file]
```

查看远程分支的变动情况。

```
$ git log remote/branch
```

查找 log，即搜索 commit信息。

```
$ git log --author=Andy
$ git log -i --grep="Something in the message"
```

上面代码中，`-i` 参数表示搜索时忽略大小写。

查看某个范围内的 commit

```
$ git log origin/master..new
# [old]..[new] - everything you haven't pushed yet
```

美化输出。

```
git log --graph --decorate --pretty=oneline --abbrev-commit
```

- --graph commit之间将展示连线
- --decorate 显示 commit里面的分支
- --pretty=oneline 只显示 commit信息的标题

- --abbrev-commit □□□ commit SHA1□□ 7□

# □□□□

# --oneline

`git log`□□□□□□ commit □□□□□□□□□□□□□ `--oneline`□□□□□□□□□ commit □□□□□□

```
$ git log --oneline --decorate
ccc3333 (HEAD, my-feature-branch) A third commit
bbb2222 A second commit
aaa1111 A first commit
9999999 (master) Old stuff on master
```

---

□□□ #2
□ □□□ □□□ 2017-07-11 21:10:39 UTC
□ □□□ □□□ 2018-01-28 20:00:17 UTC