**Git操作 - 查看历史版本**

通过git log命令可以查看详细的历史版本信息。

git log命令非常强大而好用，在复杂系统的版本管理系统中扮演着重要的角色，这也说明git log命令在日常工作当中还是非常有用的。

例如：

* 在提交了若干更新，又或者克隆了某个项目之后，你也许想回顾下提交历史。
* 如果需要回退到之前的版本，首先我们就会查看版本库的历史版本。

**1、查看详细的历史版本记录**

使用git log命令列出历史提交记录如下：

|  |
| --- |
| Bash $ git log commit 3c2eabe9de439f1cc794a1672291e95a31c84c36 (HEAD -> master) Author: sun\_wk <sun\_wk@126.com> Date: Tue Apr 13 19:04:22 2021 +0800   继续新增一行数据 v8  commit 7e7006a18f22f9f73bf2475a1b23c1403b30b32f Author: sun\_wk <sun\_wk@126.com> Date: Tue Apr 13 19:03:15 2021 +0800   test.txt 文件新增一行数据  # 以下省略... |

说明：

执行git log命令，不加任何参数的话，每一个版本记录单元的显示，均由五行构成：.

* 第1行：commit的id，由于Git是分布式版本控制系统，整个系统中存在有多个版本库，为了保证各个版本库中commit的id不重复，所有Git中的commit的id不是顺序递增的，而是与版本库，提交者，提交时间等相关的内容计算出来的一个值。
* 第2行：作者的信息。
* 第3行：提交的时间。
* 第4行：分隔行，即空行。将前面所述基本信息与后面的提交说明内容进行分隔。
* 第5行：提交说明信息。

|  |
| --- |
| 注意：git log后不加分支名称的话，是显示当前工作分支的版本历史。（分支内容后边文章详解） |

**2、简化显示历史版本记录**

常用参数如下：

**（1）列表的形式查看历史版本记录**

添加--pretty=oneline选项，以单行形式简单展示历史记录信息。

|  |
| --- |
| Bash L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log --pretty=oneline 3c2eabe9de439f1cc794a1672291e95a31c84c36 (HEAD -> master) 继续新增一行数据 v8 7e7006a18f22f9f73bf2475a1b23c1403b30b32f test.txt 文件新增一行数据 92cf82d74084977f17976e37c69681a71f64d4fc rename readme.txt -> test.txt fe20ce40a67a48e7f6b3bc3495bdde7a25dae61d rename test.txt -> readme.txt 995fbbe6ea16aceb9e14bdc959d6150d90d585d5 add test.txt file |

**（2）commit-id简写形式**

使用--pretty=oneline --abbrev-commit选项或者--oneline参数。

（abbrev：缩写，简写）

|  |
| --- |
| Bash # --pretty=oneline --abbrev-commit L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log --pretty=oneline --abbrev-commit 3c2eabe (HEAD -> master) 继续新增一行数据 v8 7e7006a test.txt 文件新增一行数据 92cf82d rename readme.txt -> test.txt fe20ce4 rename test.txt -> readme.txt 995fbbe add test.txt file  # --oneline L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log --oneline 3c2eabe (HEAD -> master) 继续新增一行数据 v8 7e7006a test.txt 文件新增一行数据 92cf82d rename readme.txt -> test.txt fe20ce4 rename test.txt -> readme.txt 995fbbe add test.txt file |

以上都是我们比较常用的查看历史版本记录的方式。

**3、历史版本记录常用操作**

**（1）指定查看最近几次提交的内容**

查看最近2次的提交内容，执行命令：$ git log -n2

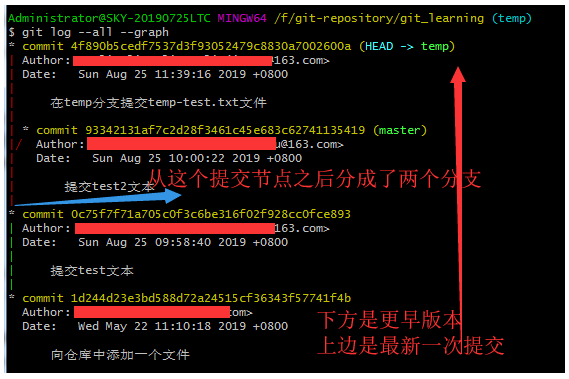
注意：-n2 可以直接写-2。

|  |
| --- |
| Bash L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log -n2 commit 3c2eabe9de439f1cc794a1672291e95a31c84c36 (HEAD -> master) Author: sun\_wk <sun\_wk@126.com> Date: Tue Apr 13 19:04:22 2021 +0800   继续新增一行数据 v8  commit 7e7006a18f22f9f73bf2475a1b23c1403b30b32f Author: sun\_wk <sun\_wk@126.com> Date: Tue Apr 13 19:03:15 2021 +0800   test.txt 文件新增一行数据 |

**（2）以简单图形的方式查看分支版本历史**

执行命令：$ git log -all --graph，可以更清楚的查看所有分支的演进历史。

如下图：



**（3）翻页与退出**

当执行git log命令显示的内容太多时，是无法在一页内显示完毕所有历史版本内容的，其最后一行会出现一个冒号，我们可以输入命令，也可以操作键盘。

常用的命令有：

* 上下键：可以看上一行或下一行。
* 回车：显示下一行。
* 空格：显示下一页。
* q键：退出git log命令显示界面。

提示：出现(END)，表示历史版本信息结束。

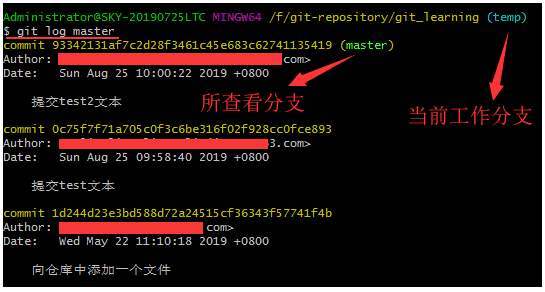
**4、查看分支相关的版本历史记录**

虽然我们还没有学习分支，这里先简单了解一下。

**（1）查看指定分支版本历史**

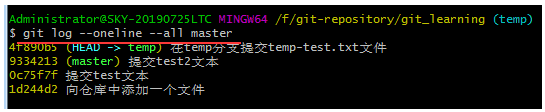
执行命令：git log + 分支名称。

我们就以master主分支为例：



**（2）查看所有分支历史版本**

执行命令：git log --all。

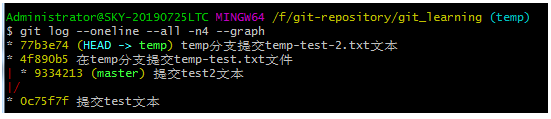


|  |
| --- |
| 注意：--all参数后在指定查看某个分支，他就不会在起作用了，结果会显示全部分支的提交日志。 |

**5、总结**

**（1）以上的几个命令可以叠加使用**

如下：



但是注意，-n4表示的是所有分支记录总共的最后4条。

**（2）如果前边用了参数--all，后边又添加了执行想看哪个分支的日志，这个执行分支是无效的，还是查看所有的分支日志。**

取掉--all参数就可以查看到指定的分支。

|  |
| --- |
| 提示：Git的指令是非常丰富的，通过git help log 或者git help --web log，可以以浏览器的方式查看更多关于log的指令。 |

**6、拓展：git blame命令**

如果要查看指定文件的修改记录可以使用git blame命令，格式如下：

|  |
| --- |
| XML git blame <file> |

git blame命令是以列表形式显示修改记录，如下示例：

|  |
| --- |
| Bash # 查看test.txt内容 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ cat test.txt hello git hello git v4 v8  # 查看test.txt历史修改记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git blame test.txt ^995fbbe (sun\_wk 2021-04-12 23:18:13 +0800 1) hello git 7e7006a1 (sun\_wk 2021-04-13 19:03:15 +0800 2) hello git v4 3c2eabe9 (sun\_wk 2021-04-13 19:04:22 +0800 3) v8 |

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/98ce9b7f2bd3 来源：简书 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。