git log常用命令以及技巧

1.git log

如果不带任何参数，它会列出所有历史记录，最近的排在最上方，显示提交对象的哈希值，作者、提交日期、和提交说明。如果记录过多，则按Page Up、Page Down、↓、↑来控制显示；按q退出历史记录列表。

2.git log -n

如果不想向上面那样全部显示，可以选择显示前N条。

3.git log --stat -n

显示简要的增改行数统计,每次提交文件的变更统计，-n 同上，前n条，可省略。

$ git log --stat -2 commit d0b9a20fac8abc7517c5a04c0fbb1d488f309bf5Author: BeginMan <pythonsuper@gmail.com> Date: Sat Mar 1 23:26:43 2014 +0800 ok \_posts/2014-02-27-Customizing-Git.md | 5 +++++ 1 file changed, 5 insertions(+) commit 8c186cd71492b7a3eae6df7de880b99efa0f87cfAuthor: BeginMan <pythonsuper@gmail.com> Date: Sat Mar 1 23:26:10 2014 +0800 mi \_posts/2014-02-27-Customizing-Git.md | 56 +++++++++++++++++++++++++++++++++++- 1 file changed, 55 insertions(+), 1 deletion(-)

每个提交都列出了修改过的文件，以及其中添加和移除的行数，并在最后列出所有增减行数小计,比如上面的有5行做了更新。

4.git log -p -n

此命令同上，不过显示更全了。

5.git log --pretty=oneline

一行显示，只显示哈希值和提交说明。

6.gig lot --graph

ASCII 字符串表示的简单图形，形象地展示了每个提交所在的分支及其分化衍合情况

$ git log --pretty=format:"%h %s" --graph

7.$ git log --pretty=format:" "

控制显示的记录格式，常用的格式占位符写法及其代表的意义如下：

选项 说明 %H 提交对象（commit）的完整哈希字串 %h 提交对象的简短哈希字串 %T 树对象（tree）的完整哈希字串 %t 树对象的简短哈希字串 %P 父对象（parent）的完整哈希字串 %p 父对象的简短哈希字串 %an 作者（author）的名字 %ae 作者的电子邮件地址 %ad 作者修订日期（可以用 -date= 选项定制格式） %ar 作者修订日期，按多久以前的方式显示 %cn 提交者(committer)的名字 %ce 提交者的电子邮件地址 %cd 提交日期 %cr 提交日期，按多久以前的方式显示 %s 提交说明

如下操作：

$ git log --pretty=format:"%h -%an,%ar : %s" -3 d0b9a20 -BeginMan,24 hours ago : ok 8c186cd -BeginMan,24 hours ago : mib2a3100 -BeginMan,24 hours ago : what?

显示了前3条的信息，简单的哈希值，作者，提交时间，提交说明。

个人感觉这个命令挺好的，为了方面使用，还是做个别名吧：

$ git config alias.logs "log --pretty=format:'%h -%an,%ar:%s'" $ git config alias.logs log --pretty=format:'%h -%an,%ar:%s' $ git logs....

8.指定路径

比如说，指定项目路径下的所有以install.md结尾的文件的提交历史：

$ git log --pretty=oneline \*install.md

只需要加上文件路径作为参数即可。

9.指定日期、关键字、作者

如两天前的提交历史：git log --since=2.days

如指定作者为"BeginMan"的所有提交:$ git log --author=BeginMan

如指定关键字为“init”的所有提交：$ git log --grep=init

如指定提交者为"Jack"的所有提交：$ git log --committer=Jack

注意作者与提交者的关系：作者是程序的修改者，提交者是代码提交人。

如指定2天前，作者为“BeginMan”的提交含有关键字'init'的前2条记录：$ git log --since=2.days --author=BeginMan --grep=init -2

注意：上面选项后面的参数可以带单双引号，如--author="BeginMan"

使用说明如下：

选项 说明

-(n) 仅显示最近的 n 条提交

--since, --after 仅显示指定时间之后的提交。

--until, --before 仅显示指定时间之前的提交。

--author 仅显示指定作者相关的提交。

--committer 仅显示指定提交者相关的提交。

来看一个实际的例子，如果要查看 Git 仓库中，2008 年 10 月期间，Junio Hamano 提交的但未合并的测试脚本（位于项目的 t/ 目录下的文件），可以用下面的查询命令：

$ git log --pretty="%h - %s" --author=gitster --since="2008-10-01" \  
--before="2008-11-01" --no-merges -- t/

[如果了解更多关于git grep可以参考这里](http://gitbook.liuhui998.com/4_8.html)

参考：

git log 命令支持的选项

-p 按补丁格式显示每个更新之间的差异。

--stat 显示每次更新的文件修改统计信息。

--shortstat 只显示 --stat 中最后的行数修改添加移除统计。

--name-only 仅在提交信息后显示已修改的文件清单。

--name-status 显示新增、修改、删除的文件清单。

--abbrev-commit 仅显示 SHA-1 的前几个字符，而非所有的 40 个字符。

--relative-date 使用较短的相对时间显示（比如，“2 weeks ago”）。

--graph 显示 ASCII 图形表示的分支合并历史。

--pretty 使用其他格式显示历史提交信息。可用的选项包括 oneline，short，full，fuller 和 format（后跟指定格式）。