## git log

**一、git log 的常用选项**

| **type** | **detail** |
| --- | --- |
| -p | 按补丁格式显示每个提交引入的差异。 |
| --stat | 显示每次提交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 校验和所有 40 个字符中的前几个字符。 |
| --relative-date | 使用较短的相对时间而不是完整格式显示日期（比如“2 weeks ago”）。 |
| --graph | 在日志旁以 ASCII 图形显示分支与合并历史。 |
| --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（用来定义自己的格式）。 |
| --oneline | --pretty=oneline --abbrev-commit 合用的简写。 |

**二、git log --pretty=format 常用的选项**

| **type** | **detail** |
| --- | --- |
| %H | 提交的完整哈希值 |
| %h | 提交的简写哈希值 |
| %T | 树的完整哈希值 |
| %t | 树的简写哈希值 |
| %P | 父提交的完整哈希值 |
| %p | 父提交的简写哈希值 |
| %an | 作者名字 |
| %ae | 作者的电子邮件地址 |
| %ad | 作者修订日期（可以用 --date=选项 来定制格式） |
| %ar | 作者修订日期，按多久以前的方式显示 |
| %cn | 提交者的名字 |
| %ce | 提交者的电子邮件地址 |
| %cd | 提交日期 |
| %cr | 提交日期（距今多长时间） |
| %s | 提交说明 |

**三、限制 git log 输出的选项**

| **type** | **detail** |  |
| --- | --- | --- |
| -<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。 | 类似 --since 和 --until 这种按照时间作限制的选项很有用。 例如，下面的命令会列出最近两周的所有提交：  $ git log --since=2.weeks  该命令可用的格式十分丰富——可以是类似 "2008-01-15" 的具体的某一天，也可以是类似 "2 years 1 day 3 minutes ago" 的相对日期 |
| --until, --before | 仅显示指定时间之前的提交。 |  |
| --author | 仅显示作者匹配指定字符串的提交。 |  |
| --committer | 仅显示提交者匹配指定字符串的提交。 |  |
| --grep | 仅显示提交说明中包含指定字符串的提交。 |  |
| -S | 仅显示添加或删除内容匹配指定字符串的提交。 | 过滤器是 -S（俗称“pickaxe”，取“用鹤嘴锄在土里捡石头”之意）， 它接受字符串参数，并且只显示添加或删除了该字符串的提交。找出添加或删除了对某个函数的引用的提交：  $ git log -S function\_name |

|  |  |
| --- | --- |
| **Note** | 用 --author 选项显示指定作者的提交，用 --grep 选项搜索提交说明中的关键字。  你可以指定多个 --author 和 --grep 搜索条件，这样会只输出 **任意** 匹配 --author 模式和 --grep 模式的提交。然而，如果你添加了 --all-match 选项， 则只会输出 **所有** 匹配 --grep 模式的提交。 |
| **Tip** | 隐藏合并提交  按照你代码仓库的工作流程，记录中可能有为数不少的合并提交，它们所包含的信息通常并不多。 为了避免显示的合并提交弄乱历史记录，可以为 log 加上 --no-merges 选项。 |