

Git Log Recap

Let's do a quick recap of the git log command. The git log command is used to display all of the commits of a repository.

```
$ git log
```

By *default*, this command displays:

- the SHA
- the author
- the date
- and the message

...of every commit in the repository. I stress the "By default" part of what Git displays because the git log command can display a lot more information than just this.

Git uses the command line pager, Less, to page through all of the information. The important keys for Less are:

- to scroll down by a line, use j or ↓
- to scroll up by a line, use k or ↑
- to scroll down by a page, use the spacebar or the Page Down button
- to scroll up by a page, use b or the Page Up button
- to quit, use q

git log --oneline Recap

To recap, the --oneline flag is used to alter how git log displays information:

```
$ git log --oneline
```

This command:

- lists one commit per line
- shows the first 7 characters of the commit's SHA
- shows the commit's message

git log --stat Recap

To recap, the --stat flag is used to alter how git log displays information:

```
$ git log --stat
```

This command:

- displays the file(s) that have been modified
- displays the number of lines that have been added/removed
- displays a summary line with the total number of modified files and lines that have been added/removed

git log -p Recap

To recap, the -p flag (which is the same as the --patch flag) is used to alter how git log displays information:

```
$ git log -p
```

This command adds the following to the default output:

- displays the files that have been modified
- displays the location of the lines that have been added/removed
- displays the actual changes that have been made

Viewing A Specific Commit

But did you know, you can supply the SHA of a commit as the final argument for all of these commands? For example:

```
$ git log -p <SHA>
```

```
$ git log -p fdf5493
```

By supplying a SHA, the git log -p command will *start at that commit*! No need to scroll through everything! Keep in mind that it will *also* show all of the commits that were made *prior* to the supplied SHA.

New Command: git show

The other command that shows a specific commit is git show:

```
$ git show
```

Running it like the example above will only display the most recent commit. Typically, a SHA is provided as a final argument:

```
$ git show fdf5493
```

What does git show do?

The git show command will show *only one commit*. So don't get alarmed when you can't find any other commits - it only shows one. The output of the git show command is exactly the same as the git log -p command. So by default, git show displays:

- the commit
- the author
- the date
- the commit message
- the patch information

However, git show can be combined with most of the other flags we've looked at:

- --stat - to show the how many files were changed and the number of lines that were added/removed
- -p or --patch - this the default, but if --stat is used, the patch won't display, so pass -p to add it again
- -w - to ignore changes to whitespace