Tuesday 02/10/2015

Very useful Git Bash Commands for log:

(1) Viewing the Commit History

```
$ git log
```

=======

(2) compare certain commit diff

```
$ git log -p -2
```

One of the more helpful options is -p, which shows the difference introduced in each commit. You can also use -2, which limits the output to only the last two entries:

==========

(3) summary of commits

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

You can also use a series of summarizing options with git log. For example, if you want to see some abbreviated stats for each commit, you can use the --stat option:

==============

(4) display log in better format

```
$ git log --pretty=oneline
```

Another really useful option is --pretty. This option changes the log output to formats other than the default. A few prebuilt options are available for you to use. The oneline option prints each commit on a single line, which is useful if you're looking at a lot of commits. In addition, the short, full, and fuller options show the output in roughly the same format but with less or more information, respectively:

==============

(5) display more detailed info (i.e. uname)

```
git log --pretty=format:"%h - %an, %ar : %s"
```

The most interesting option is format, which allows you to specify your own log output format. This is especially useful when you're generating output for machine parsing – because you specify the format explicitly, you know it won't change with updates to Git:

Useful options for git log --pretty=format

Option Description of Output

%H

Commit hash

%h

Abbreviated commit hash

%T

Tree hash

%t

Abbreviated tree hash

%P

Parent hashes

%p

Abbreviated parent hashes

%an

Author name

%ae

Author e-mail

%ad

Author date (format respects the --date=option)

%ar

Author date, relative

%cn

Committer name

%ce

Committer email

%cd

Committer date

%cr

Committer date, relative

%s

Subject

==================

(6) graph

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

The oneline and format options are particularly useful with another log option called --graph. This option adds a nice little ASCII graph showing your branch and merge history:

===================

(7) Limiting Log Output

```
git log --since=2.weeks
```

However, the time-limiting options such as --since and --until are very useful. For example, this command gets the list of commits made in the last two weeks:

This command works with lots of formats – you can specify a specific date like "2008-01-15", or a relative date such as "2 years 1 day 3 minutes ago".

=======================

(8) change happened to ceratin function

```
$ git log -Sfunction_name
```

=================

(9) complicated log command

```
$ git log --pretty="%h - %s" --author=gitster --since="2008-10-01" \
```