

10. HISTORY

Goals

- To learn to view the project's history.

Getting a list of changes made is a function of the `git log` command.

RUN:

```
git log
```

You will see ...

RESULT:

```
$ git log
commit fa3c1411aa09441695a9e645d4371e8d749da1dc
Author: Marina Pushkova <marina (at) githowto.com>
Date:   Wed Mar 9 10:27:54 2011 -0500
```

Added HTML header

```
commit 8c3228730ed03116815a5cc682e8105e7d981928
Author: Marina Pushkova <marina (at) githowto.com>
Date:   Wed Mar 9 10:27:54 2011 -0500
```

Added standard HTML page tags

```
commit 43628f779cb333dd30d78186499f93638107f70b
Author: Marina Pushkova <marina (at) githowto.com>
Date:   Wed Mar 9 10:27:54 2011 -0500
```

Added h1 tag

```
commit 911e8c91caeab8d30ad16d56746cbd6eef72dc4c
Author: Marina Pushkova <marina (at) githowto.com>
Date:   Wed Mar 9 10:27:54 2011 -0500
```

First Commit

Here is a list of all the four commits to the repository, which we were able to make so far.

One line history

You fully control over what the log shows. I like the single line format:

RUN:

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

You will see ...

RESULT:

```
$ git log --pretty=oneline
fa3c1411aa09441695a9e645d4371e8d749da1dc Added HTML header
8c3228730ed03116815a5cc682e8105e7d981928 Added standard HTML page tags
43628f779cb333dd30d78186499f93638107f70b Added h1 tag
911e8c91caeab8d30ad16d56746cbd6eef72dc4c First Commit
```

Controlling the display of entries

There are many options to choose which entries appear in the log. Play around with the following parameters:

```
git log --pretty=oneline --max-count=2
git log --pretty=oneline --since='5 minutes ago'
git log --pretty=oneline --until='5 minutes ago'
git log --pretty=oneline --author=<your name>
git log --pretty=oneline --all
```

Details are provided in the `git-log` instruction.

Getting fancy

This is what I use to review the changes made within the last week. I will add `--author=marina` if I want to see only the changes made by me.

```
git log --all --pretty=format:"%h %cd %s (%an)" --since='7 days ago'
```

The ultimate format of the log

Over time, I found the following log format to be the most suitable.

RUN:

```
git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
```

It looks like this:

RESULT:

```
$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* fa3c141 2011-03-09 | Added HTML header (HEAD, master) [Marina Pushkova]
* 8c32287 2011-03-09 | Added standard HTML page tags [Marina Pushkova]
* 43628f7 2011-03-09 | Added h1 tag [Marina Pushkova]
* 911e8c9 2011-03-09 | First Commit [Marina Pushkova]
```

Let's look at it in detail:

- `--pretty="..."` defines the output format.
- `%h` is the abbreviated hash of the commit
- `%d` commit decorations (e.g. branch heads or tags)
- `%ad` is the commit date
- `%s` is the comment
- `%an` is the name of the author
- `--graph` tells git to display the commit tree in the form of an ASCII graph layout
- `--date=short` keeps the date format short and nice

So, every time you want to see a log, you'll have to do a lot of typing. Fortunately, we will find out about the git aliases in the next lesson.

Other tools

Both `gitx` (for Mac) and `gitk` (for any platform) can help to explore log history.

◀ 9. Changes, not files

11. Aliases ▶

We plan to roll-out a major upgrade to this tutorial pretty soon. We can keep you posted if you want.

Your email:

Subscribe