Viewing the Commit History

coursera.org/learn/git-distributed-development/supplement/5feAU/viewing-the-commit-history

It is easy to display the history of commits with git using the command **git log**. For example, consider the following script which sets up a repository and then adds some files, modifies them and introduces four commits along the way:

```
#!/bin/bash

rm -rf git-test
mkdir git-test

cd git-test
git init

git config user.name "A Smart Guy"
git config user.email "asmartguy@linux.com"

echo file1 > file1
git add file1
git commit file1 -m "This is the first commit"

echo file2 > file2
git add file2
git commit . -m "This is the second commit"

echo file3 > file3
echo another line for file3 >> file3
git add .
git commit . -m "This is the third commit"
```

```
echo another line for file2 >> file2

git add .

git commit -a  -m "This is the fourth commit"
```

If we then ask to see the log, we see:

```
$ git log

commit 4b4bf2c5aa95b6746f56f9dfce0e4ec6bddad407
Author: A Smart Guy <asmartguy@linux.com>
Date:   Thu Dec 31 13:50:15 2009 -0600

    This is the fourth commit

commit 55eceacc9ab2b4fc1c806b26e79eca4429d8b52a
Author: A Smart Guy <asmartguy@linux.com>
Date:   Thu Dec 31 13:50:15 2009 -0600

    This is the third commit

commit f60c0c21764676beca75b7edc2f5f5e51b5dd404
Author: A Smart Guy <asmartguy@linux.com>
Date:   Thu Dec 31 13:50:15 2009 -0600

    This is the second commit

commit 712cbafa7ee0aaef03861b049ddc7865220b4e2c
Author: A Smart Guy <asmartguy@linux.com>
Date:   Thu Dec 31 13:50:15 2009 -0600
```

```
    This is the first commit
```

The commits are shown in reverse order of introduction. Even shorter, you can do:

```
$ git log --pretty=oneline

4b4bf2c5aa95b6746f56f9dfce0e4ec6bddad407 This is the fourth commit
55eceacc9ab2b4fc1c806b26e79eca4429d8b52a This is the third commit
f60c0c21764676beca75b7edc2f5f5e51b5dd404 This is the second commit
712cbafa7ee0aaef03861b049ddc7865220b4e2c This is the first commit
```

You can also see the actual patches made with the **-p** option and can view only part of the history by specifying a particular commit, as in:

```
$ git log -p f60c

commit f60c0c21764676beca75b7edc2f5f5e51b5dd404
Author: A Smart Guy <asmartguy@linux.com>
Date:   Thu Dec 31 13:50:15 2009 -0600

    This is the second commit
```

```
diff --git a/file2 b/file2
new file mode 100644
index 0000000..6c493ff
--- /dev/null
+++ b/file2
@@ -0,0 +1 @@
+file2

commit 712cbafa7ee0aaef03861b049ddc7865220b4e2c
Author: A Smart Guy <asmartguy@linux.com>
Date:   Thu Dec 31 13:50:15 2009 -0600

    This is the first commit

diff --git a/file1 b/file1
new file mode 100644
index 0000000..e212970
--- /dev/null
+++ b/file1
@@ -0,0 +1 @@
+file1
```

There are a lot of other options to **git log** which you can view by doing either **git help** log or **man git log**.