Short Introduction to git

- why to use version control
- procedure for using `git`
- basic commands for `git`
- things to explore by yourself

Why to use “version control”?

- it's a way to keep track of modifications to files
    - source code
    - scripts
    - ...
- may help to avoid directories like `oldStuff`, `test1-new`, `version28,...`
- each change has a comment and a timestamp

Procedure

- initialize once
- after each significant change, run
  `git commit ...`
  to write down that there is a new version
- use `git diff` to see what has changed

Basic commands – help

- `git help`
  - gives a explanation of many commands
- websites:
  - http://www.kernel.org/pub/software/scm/git/docs/everyday.html
  - http://www.kernel.org/pub/software/scm/git/docs/user-manual.html

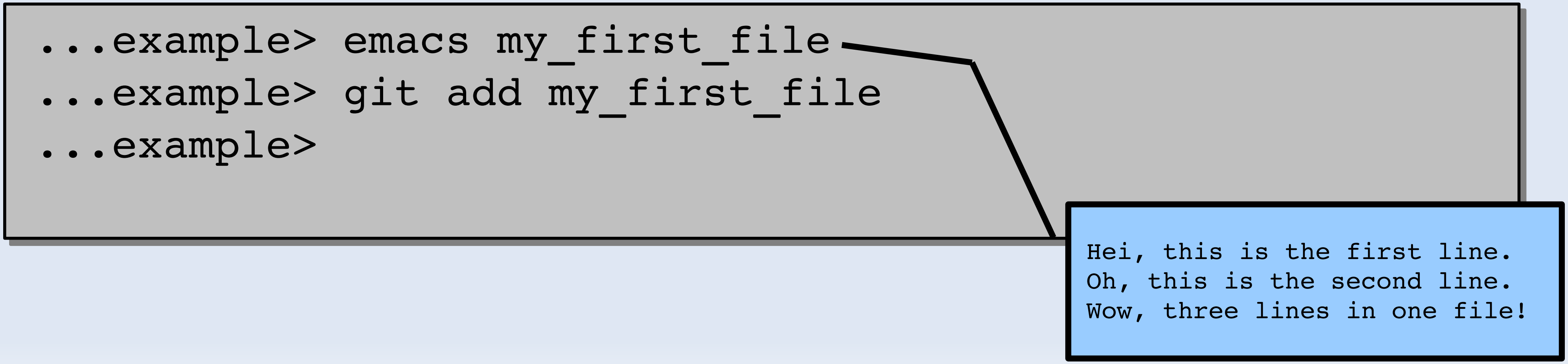
Basic commands -- init

- git init
  - initializes the database
    - creates a directory .git – better not to touch it by hand

```
...example> git init
Initialized empty Git repository in .git/
...example>
```

Basic commands -- add

- `git add filename`
  - adds a single file
- `git add directoryname`
  - adds a directory and all files in it (e.g. `git add .`)
  - use `.gitignore` to exclude some files (next slide)



```
...example> emacs my_first_file
...example> git add my_first_file
...example>
```

```
Hei, this is the first line.
Oh, this is the second line.
Wow, three lines in one file!
```

.gitignore

- used to make git ignore some files
  - especially generated files, e.g. compiler output

```
...example> more .gitignore
.gitignore
*.ignored
*.o
lib*.a
prog/my_first_program
...example>
```

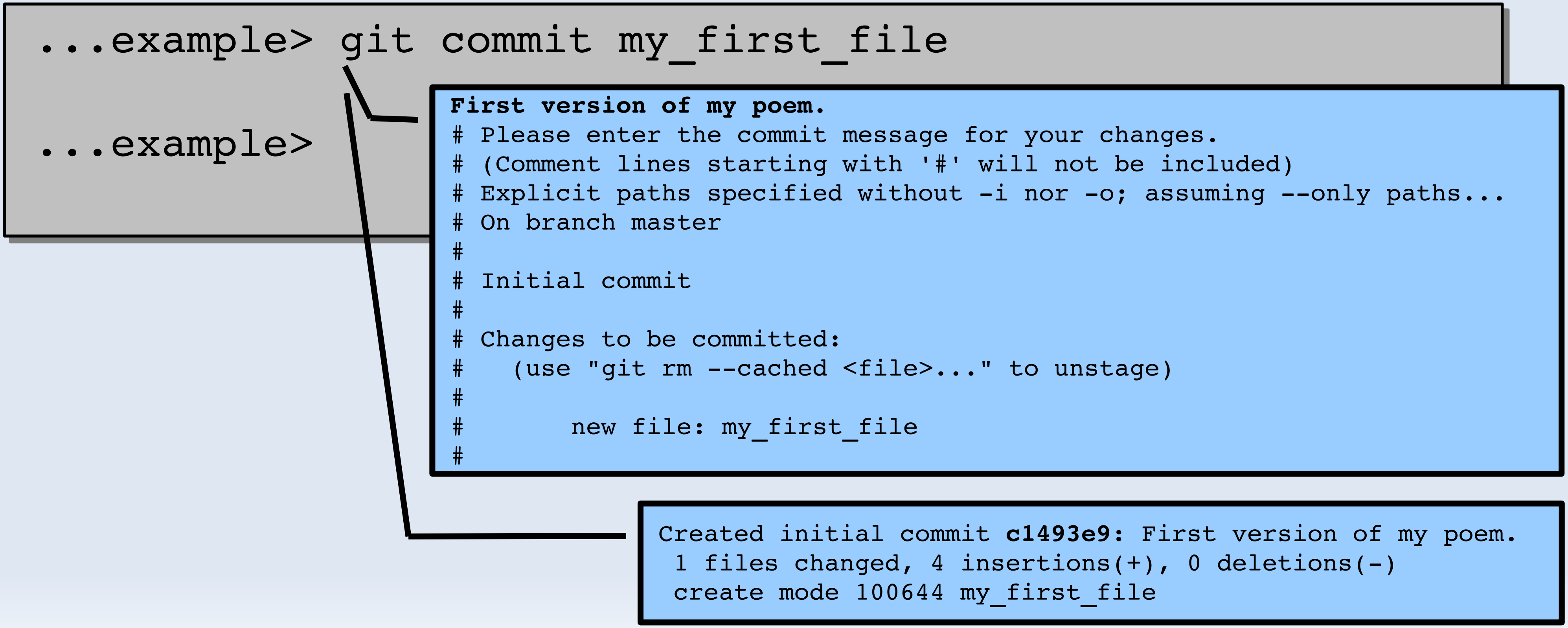
Basic commands -- status

- `git status`
  - shows changes since last commit
    - removals, renames, additions, changes

```
...example> git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file: my_first_file
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       file.notignored
#
...example>
```

Basic commands -- commit

- `git commit filename`
  - memorizes the current version of the file
- `git commit -a`
  - memorizes the current version of all (changed) files



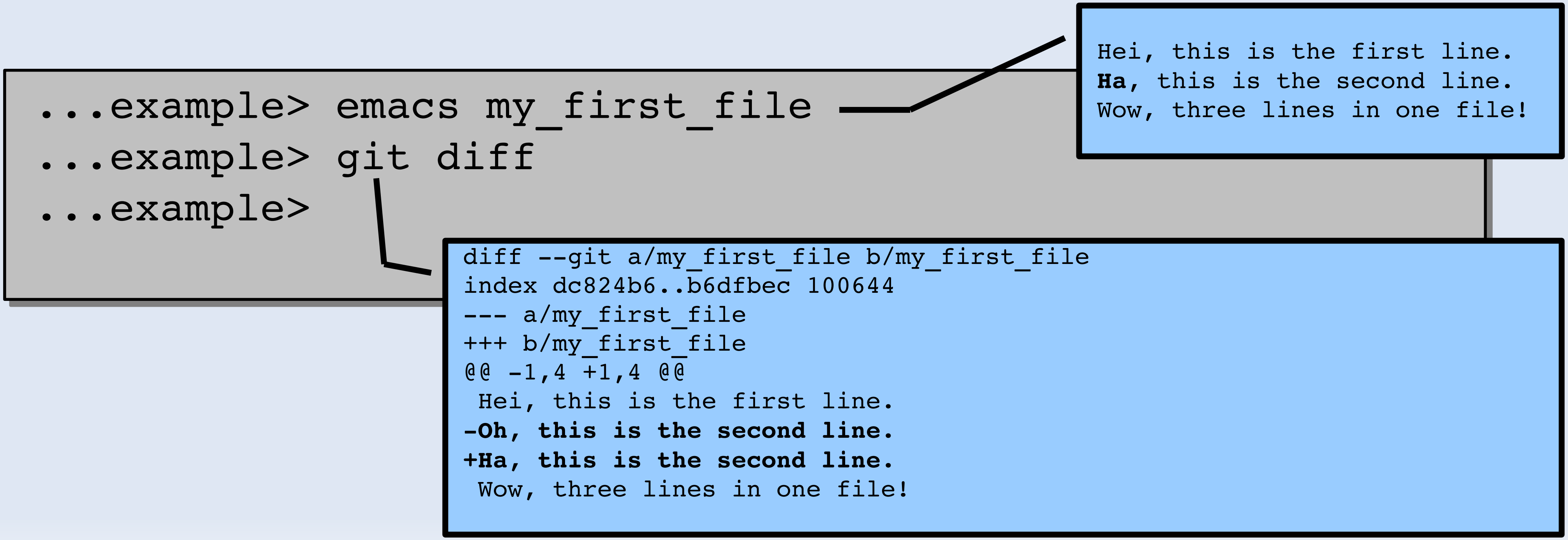
```
...example> git commit my_first_file

...example>
```

```
First version of my poem.
# Please enter the commit message for your changes.
# (Comment lines starting with '#' will not be included)
# Explicit paths specified without -i nor -o; assuming --only paths...
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file: my_first_file
#
```

```
Created initial commit c1493e9: First version of my poem.
 1 files changed, 4 insertions(+), 0 deletions(-)
 create mode 100644 my_first_file
```

Basic commands -- diff

- `git diff`
  - shows changes since last commit for all files
- `git diff filename`
  - shows changes since last commit



```
...example> emacs my_first_file
...example> git diff
...example>
```

```
Hei, this is the first line.
Ha, this is the second line.
Wow, three lines in one file!
```

```
diff --git a/my_first_file b/my_first_file
index dc824b6..b6dfbec 100644
--- a/my_first_file
+++ b/my_first_file
@@ -1,4 +1,4 @@
 Hei, this is the first line.
-Oh, this is the second line.
+Ha, this is the second line.
 Wow, three lines in one file!
```

Basic commands – mv

- `git mv oldname newname`
  - renames a file
    - the name change will be remembered

```
...example> git mv my_first_file my_first_poem
...example> git status
...example> git commit my_first_poem
...example>
```

```
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       renamed:    my_first_file -> my_first_poem
#
```

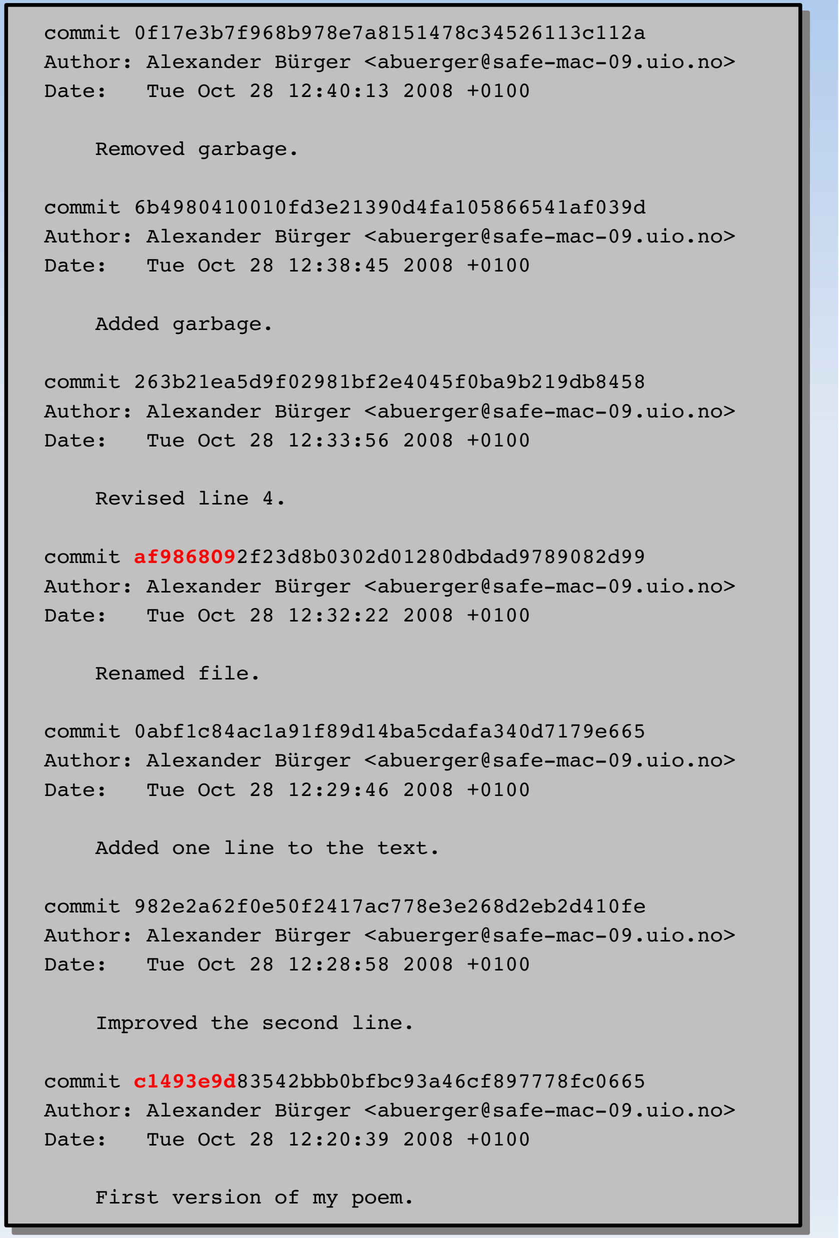
Basic commands – rm

- `git rm filename`
  - tells that the file is not needed any more
    - old versions will be remembered

```
[add a file 'my_garbage']
...example> git rm my_garbage
rm 'my_garbage'
...example> git commit my_garbage
Created commit 0f17e3b: Removed garbage.
 1 files changed, 0 insertions(+), 1 deletions(-)
 delete mode 100644 my_garbage
...example>
```

Basic commands -- log

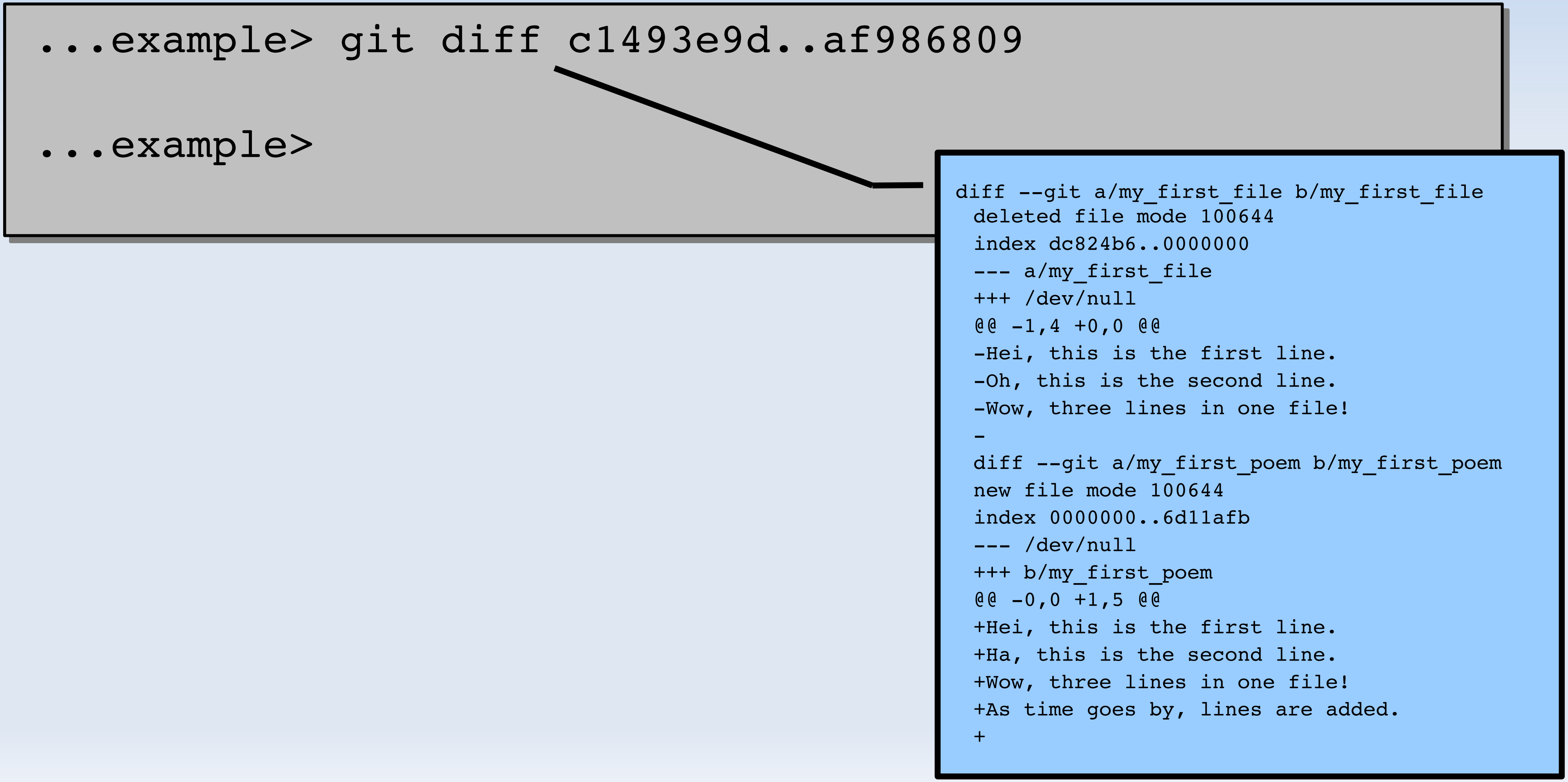
- `git log`
  - shows history
- `git log filename`
  - shows history for one file (that exists)



```
commit 0f17e3b7f968b978e7a8151478c34526113c112a
Author: Alexander Bürger <abuerger@safe-mac-09.uio.no>
Date:   Tue Oct 28 12:40:13 2008 +0100

    Removed garbage.

commit 6b4980410010fd3e21390d4fa105866541af039d
Author: Alexander Bürger <abuerger@safe-mac-09.uio.no>
Date:   Tue Oct 28 12:38:45 2008 +0100

    Added garbage.

commit 263b21ea5d9f02981bf2e4045f0ba9b219db8458
Author: Alexander Bürger <abuerger@safe-mac-09.uio.no>
Date:   Tue Oct 28 12:33:56 2008 +0100

    Revised line 4.

commit af9868092f23d8b0302d01280dbdad9789082d99
Author: Alexander Bürger <abuerger@safe-mac-09.uio.no>
Date:   Tue Oct 28 12:32:22 2008 +0100

    Renamed file.

commit 0abf1c84ac1a91f89d14ba5cdafa340d7179e665
Author: Alexander Bürger <abuerger@safe-mac-09.uio.no>
Date:   Tue Oct 28 12:29:46 2008 +0100

    Added one line to the text.

commit 982e2a62f0e50f2417ac778e3e268d2eb2d410fe
Author: Alexander Bürger <abuerger@safe-mac-09.uio.no>
Date:   Tue Oct 28 12:28:58 2008 +0100

    Improved the second line.

commit c1493e9d83542bbb0bfbc93a46cf897778fc0665
Author: Alexander Bürger <abuerger@safe-mac-09.uio.no>
Date:   Tue Oct 28 12:20:39 2008 +0100

    First version of my poem.
```

Basic commands – diff (2)

- `git diff version..version`
  - shows difference between two old versions



```
...example> git diff c1493e9d..af986809

...example>
```

```
diff --git a/my_first_file b/my_first_file
 deleted file mode 100644
 index dc824b6..0000000
 --- a/my_first_file
 +++ /dev/null
 @@ -1,4 +0,0 @@
 -Hei, this is the first line.
 -Oh, this is the second line.
 -Wow, three lines in one file!
 -
 diff --git a/my_first_poem b/my_first_poem
 new file mode 100644
 index 0000000..6d11afb
 --- /dev/null
 +++ b/my_first_poem
 @@ -0,0 +1,5 @@
 +Hei, this is the first line.
 +Ha, this is the second line.
 +Wow, three lines in one file!
 +As time goes by, lines are added.
 +
```

Basic commands -- checkout

- `git checkout filename`
  - recalls last remembered version of a file
    - e.g. if it was lost or messed up

```
...example> rm my_first_poem
...example> git checkout my_first_poem
...example> more my_first_poem
Hei, this is the first line.
Ha, this is the second line.
Wow, three lines in one file!
As time goes by, more lines are added.
```

Basic commands -- archive

- `git archive [--prefix=prefix] version`
  - produces a tar file with the state of this version
  - with prefix option, make a directory in the archive
  - see git help archive

```
...example> git archive --prefix=oldStuff/ 6b498041 | tar tf -
oldStuff/
oldStuff/my_first_poem
oldStuff/my_garbage
...example> git archive 6b498041 | gzip > oldStuff.tar.gz
...example>
```

More to Explore...

- branches
  - keep several versions in parallel
- collaboration
  - how to get changes made by somebody else
- see websites:
  - http://www.kernel.org/pub/software/scm/git/docs/everyday.html
  - http://www.kernel.org/pub/software/scm/git/docs/user-manual.html