

<https://github.com/universeworkshops/deep-clean-your-git-repositories>

wifi: github / pw: workshops2018



DEEP CLEAN YOUR GIT REPOSITORIES

@hectorsector - Hector Alfaro

@larsxschneider - Lars Schneider



WHY CLEANING?

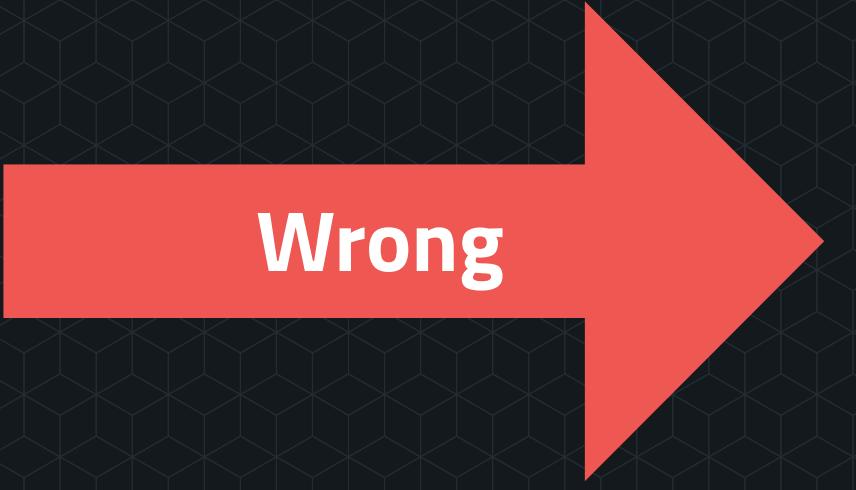


Source: <https://commons.wikimedia.org/wiki/File:Waiting.....gif>

Reduce repository size.

Reduce clone time.

Speed up Git operations.



Wrong

- Commit with valid email address**



larsxschneider committed a minute ago

Commit with invalid email address



Lras Schneider committed 5 minutes ago

Fix commit author attribution.



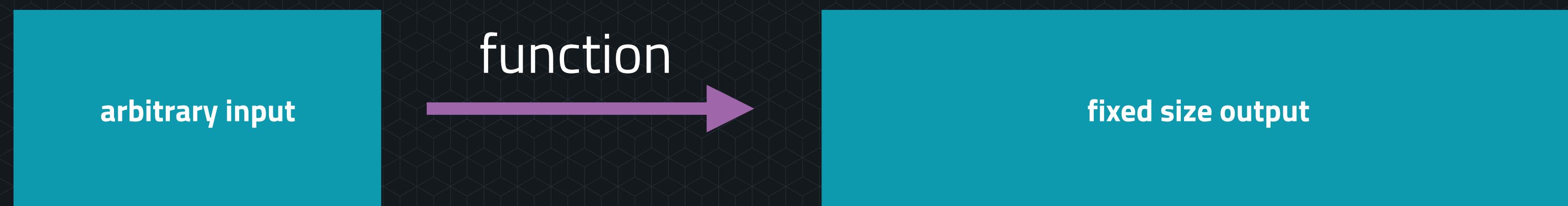
WHY IS CLEANING "COMPLICATED"?

DANGER ZONE

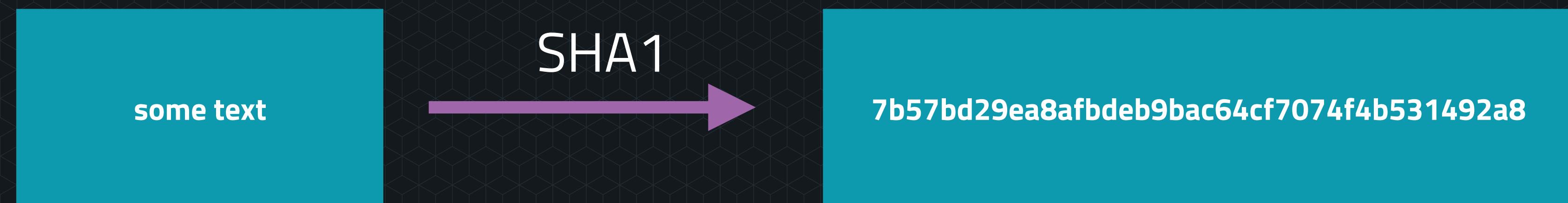


A LITTLE BIT OF THEORY

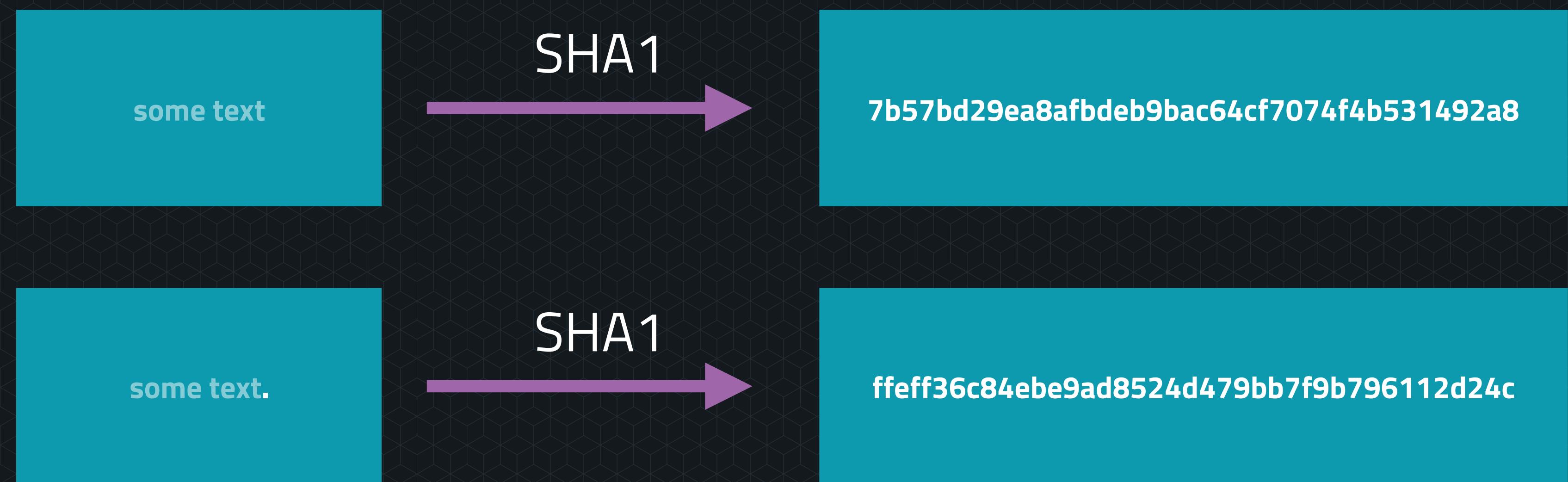
WHAT IS A HASH?



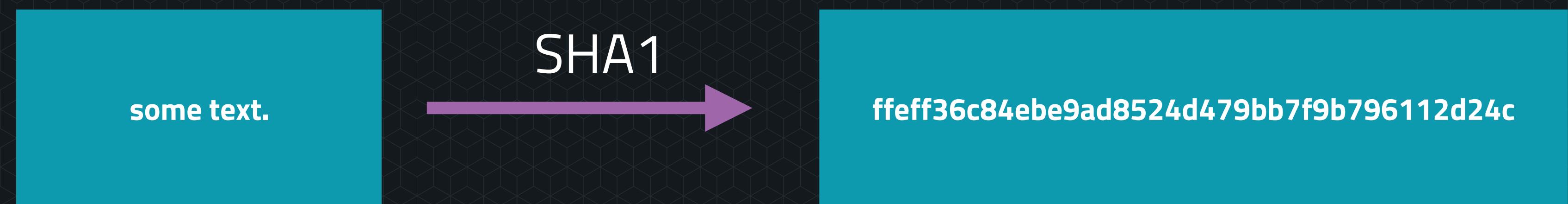
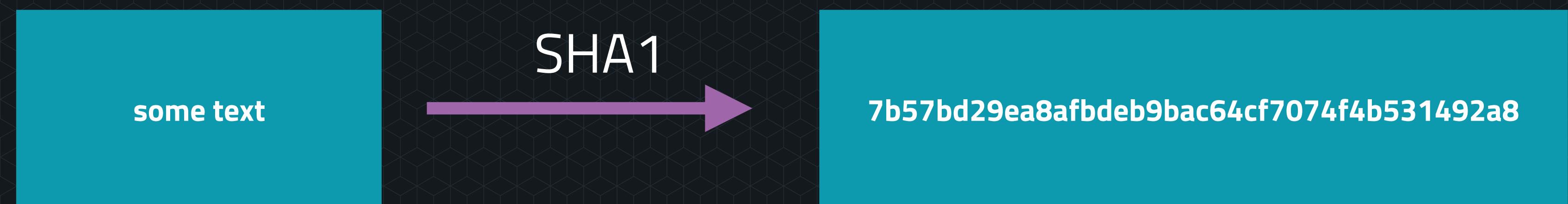
WHAT IS A HASH?



WHAT IS A HASH?



WHAT IS A HASH?



Action

```
echo 'some text' | git hash-object --stdin
```

WHAT IS A COMMIT?



WHAT IS A COMMIT?

Commit message:
My first commit



WHAT IS A COMMIT?

Commit message:

My first commit

Author:

Lras Schneider

WHAT IS A COMMIT?

Commit message:

My first commit

Author:

Lras Schneider

File list:

/tool.exe

/main.c

WHAT IS A COMMIT?



Commit message:

My first commit

Author:

Lras Schneider

File list with hashes:

/tool.exe - a1

/main.c - b1

SHA1

b1

a1

blobs

commits



SHA1

Commit message:

My first commit

Author:

Lras Schneider

File list with hashes:

/tool.exe - a1

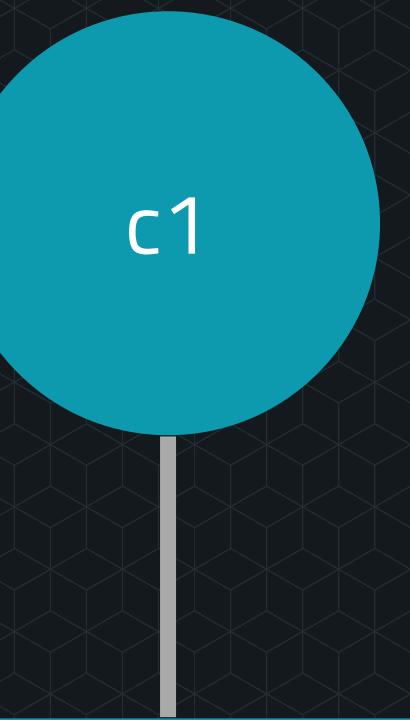
/main.c - b1

b1

a1

blobs

commits



Commit message:

My first commit

Author:

Lras Schneider

File list with hashes:

/tool.exe - **a1**

/main.c - **b1**

Commit message:

Improve main

Author:

Hector Alfaro

File list with hashes:

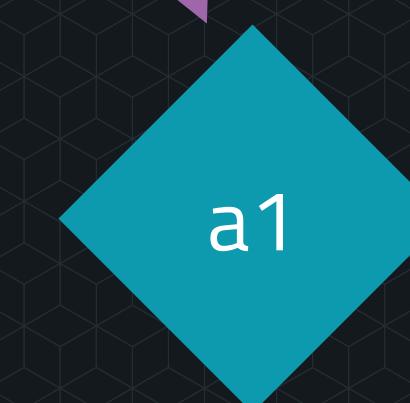
/tool.exe - **a1**

/main.c - **b2**

Parent commit hash:

c1

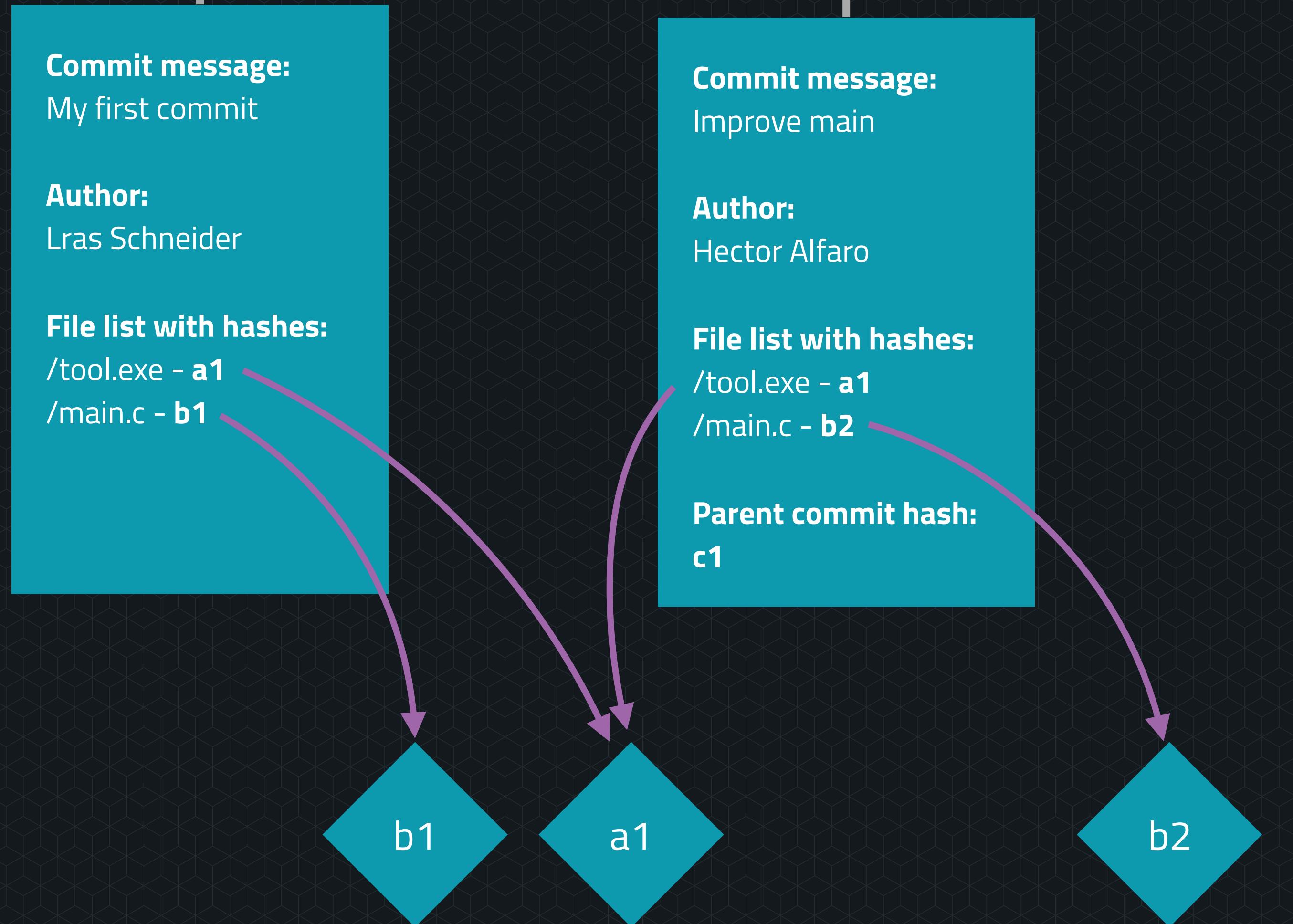
blobs



commits

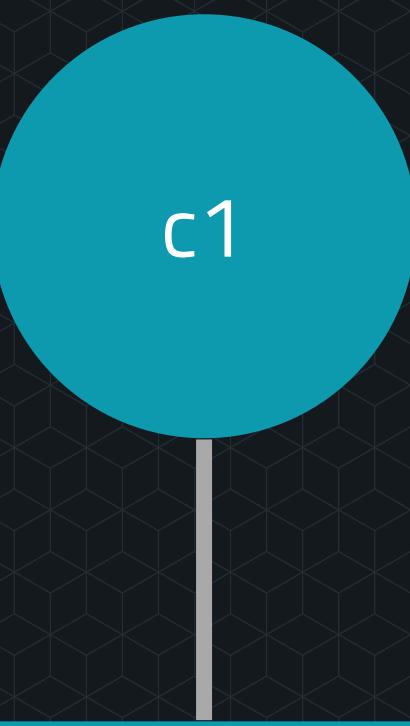
c1

c2



blobs

commits

**Commit message:**

My first commit

Author:

Lras Schneider

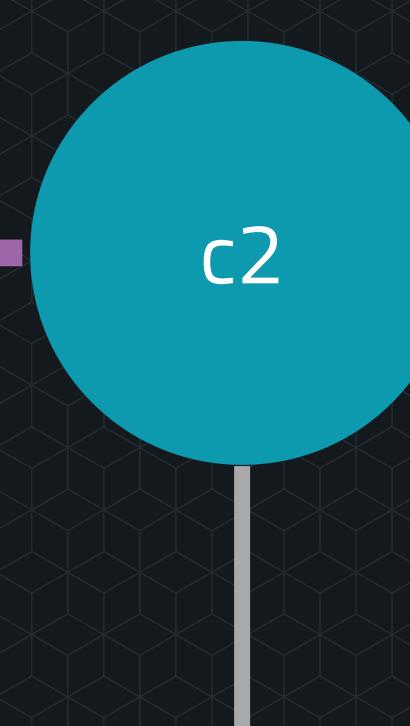
File list with hashes:

/tool.exe - **a1**

/main.c - **b1**

b1

a1

**Commit message:**

Improve main

Author:

Hector Alfaro

File list with hashes:

/tool.exe - **a1**

/main.c - **b2**

Parent commit hash:

c1

b2

**Commit message:**

Delete binary

Author:

Lars Schneider

File list with hashes:

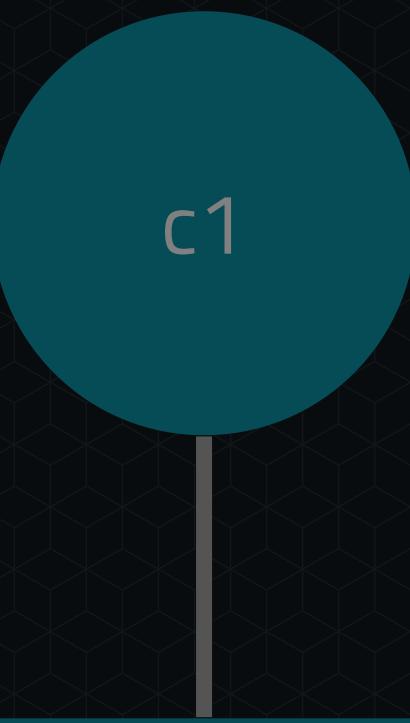
/main.c - **b2**

Parent commit hash:

c2

blobs

commits



Commit message:

My first commit

Author:

Lras Schneider

File list with hashes:

/tool.exe - a1

/main.c - b1



Commit message:

Improve main

Author:

Hector Alfaro

File list with hashes:

/tool.exe - a1

/main.c - b2

Parent commit hash:

c1



Commit message:

Delete binary

Author:

Lars Schneider

File list with hashes:

/main.c - b2



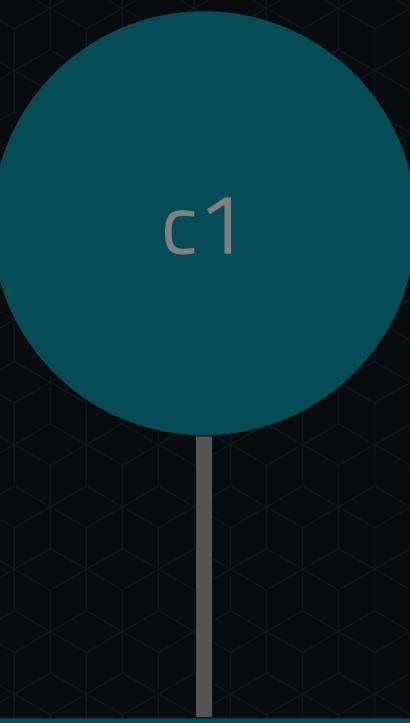
Parent commit hash:

c2

The only "visible" content!

blobs

commits



Commit message:

My first commit

Author:

Lras Schneider

File list with hashes:

/tool.exe - **a1**

/main.c - **b1**



Commit message:

Improve main

Author:

Hector Alfaro

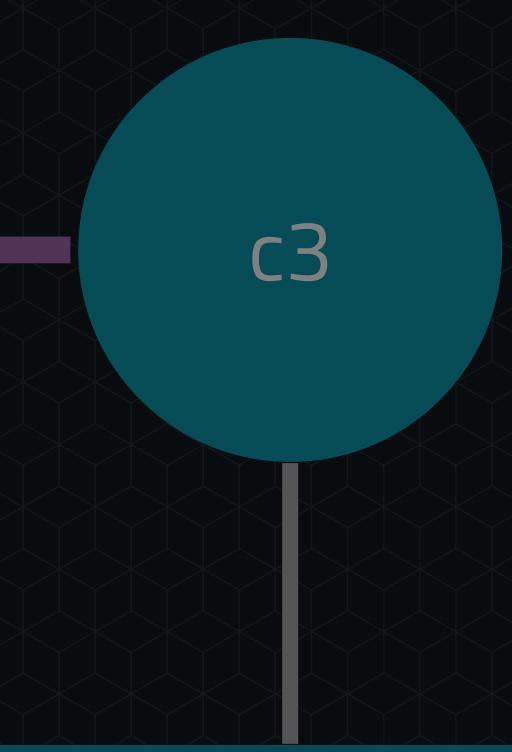
File list with hashes:

/tool.exe - **a1**

/main.c - **b2**

Parent commit hash:

c1



Commit message:

Delete binary

Author:

Lars Schneider

File list with hashes:

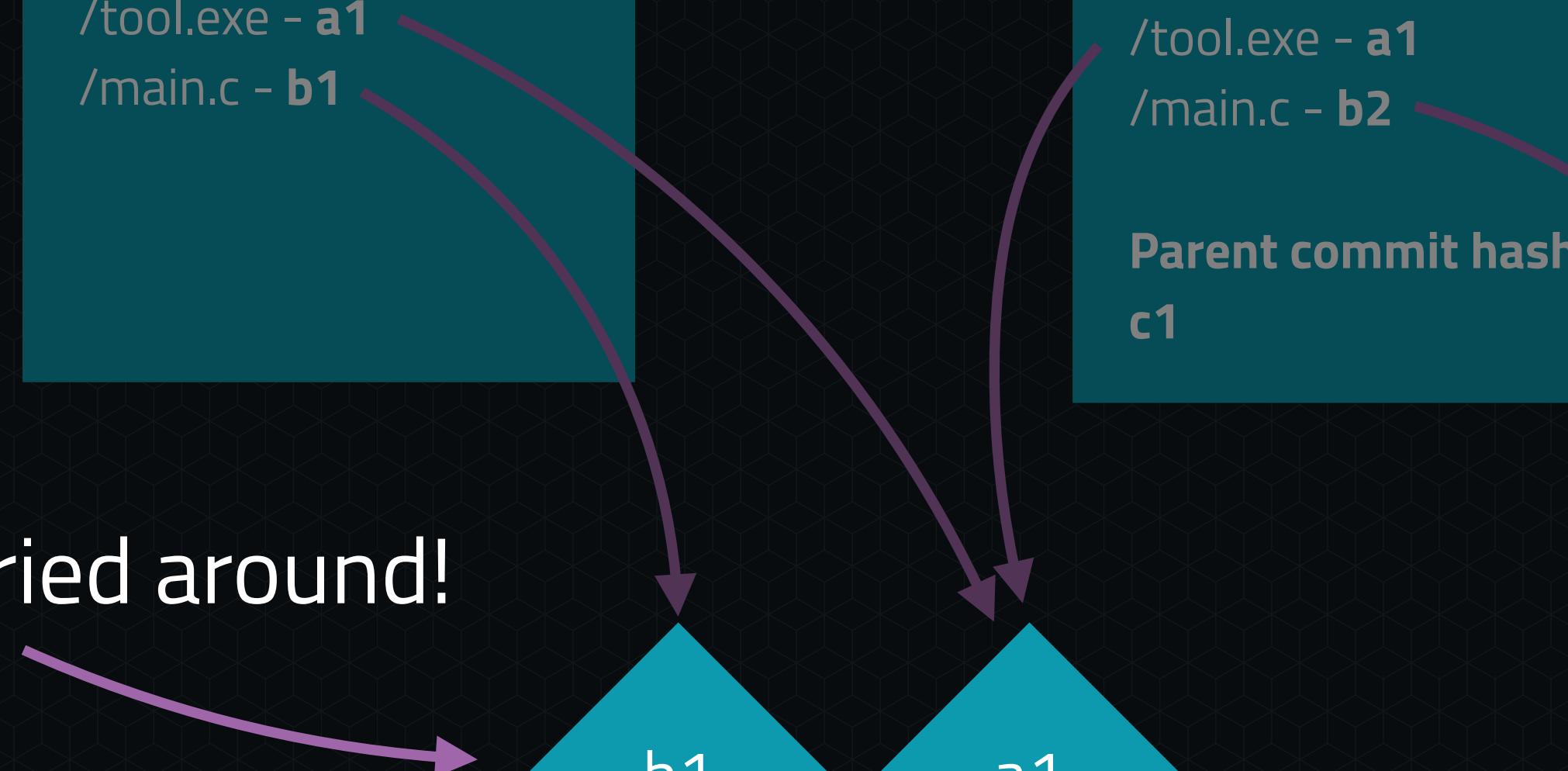
/main.c - **b2**

Parent commit hash:

c2

Still carried around!

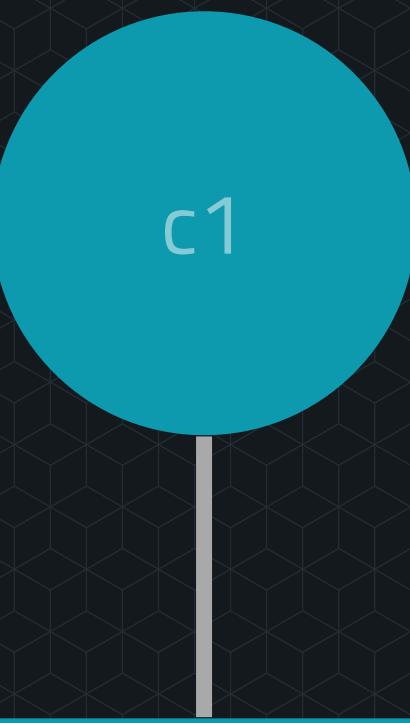
blobs





WHAT HAPPENS...
...IF YOU PURGE A FILE?

commits



Commit message:

My first commit

Author:

Lras Schneider

File list with hashes:

/tool.exe - **a1**

/main.c - **b1**

b1

a1



Commit message:

Improve main

Author:

Hector Alfaro

File list with hashes:

/tool.exe - **a1**

/main.c - **b2**

Parent commit hash:

c1

b2



Commit message:

Delete binary

Author:

Lars Schneider

File list with hashes:

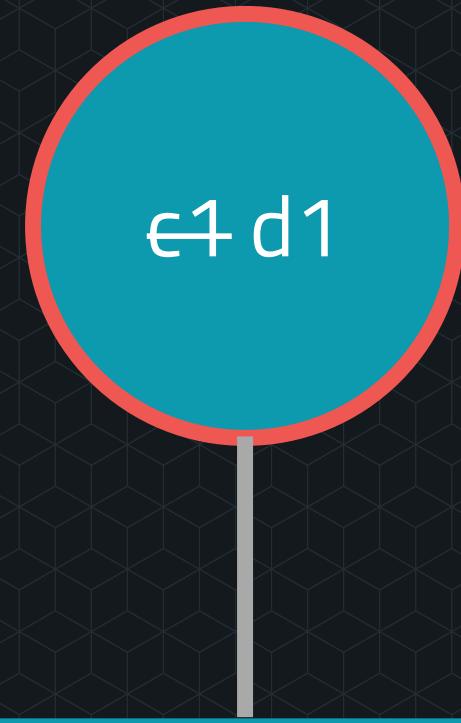
/main.c - **b2**

Parent commit hash:

c2

blobs

commits



Commit message:

My first commit

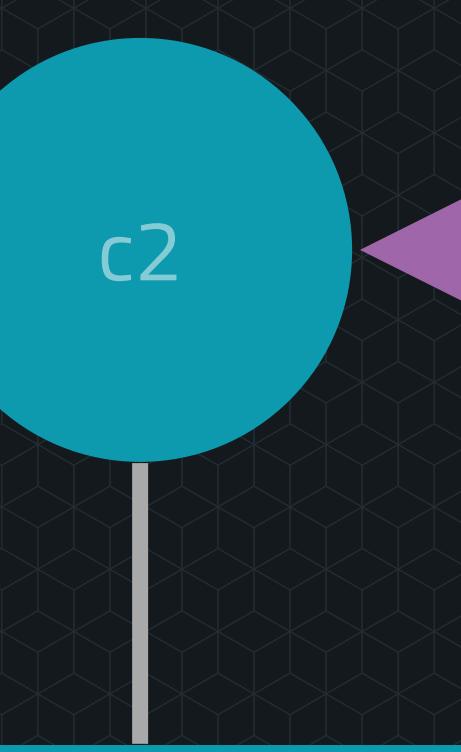
Author:

Lras Schneider

File list with hashes:

/tool.exe - **a1**

/main.c - **b1**



Commit message:

Improve main

Author:

Hector Alfaro

File list with hashes:

/tool.exe - **a1**

/main.c - **b2**

Parent commit hash:

`c1`



Commit message:

Delete binary

Author:

Lars Schneider

File list with hashes:

/main.c - **b2**

Parent commit hash:

`c2`

blobs



commits

c1 d1

c2

c3

Commit message:
My first commit

Author:
Lras Schneider

File list with hashes:
`/tool.exe - a1`
`/main.c - b1`

Commit message:
Improve main

Author:
Hector Alfaro

File list with hashes:
`/tool.exe - a1`
`/main.c - b2`

Parent commit hash:
c1 d1

Commit message:
Delete binary

Author:
Lars Schneider

File list with hashes:
`/main.c - b2`

Parent commit hash:
c2

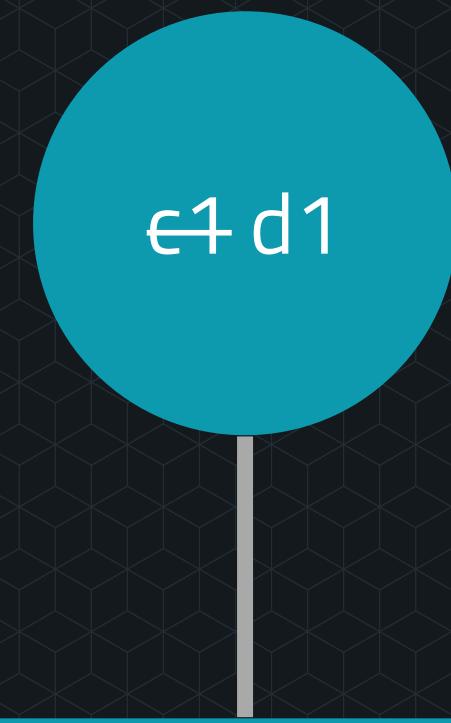
blobs

b1

a1

b2

commits



Commit message:

My first commit

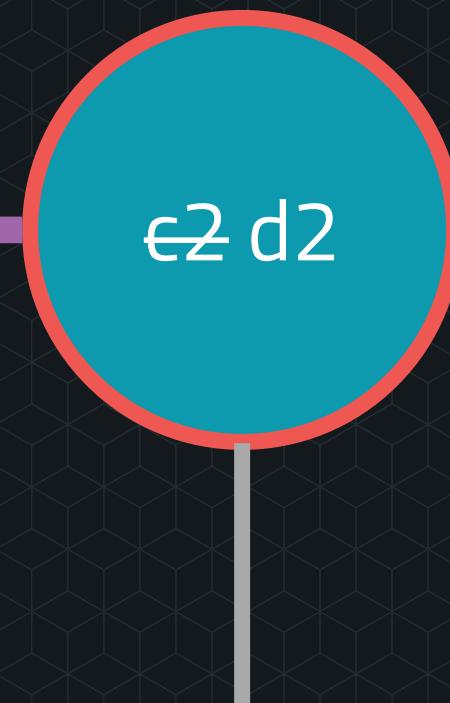
Author:

Lras Schneider

File list with hashes:

/tool.exe - **a1**

/main.c - **b1**



Commit message:

Improve main

Author:

Hector Alfaro

File list with hashes:

/tool.exe - **a1**

/main.c - **b2**

Parent commit hash:

$e1 d1$



Commit message:

Delete binary

Author:

Lars Schneider

File list with hashes:

/main.c - **b2**

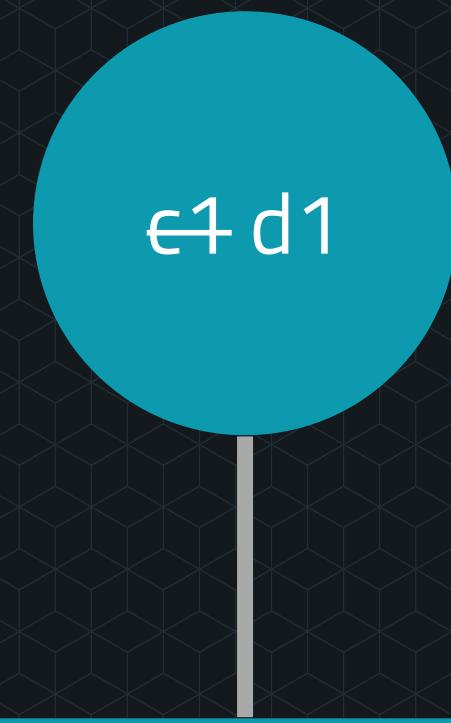
Parent commit hash:

$c2$

blobs



commits



Commit message:

My first commit

Author:

Lras Schneider

File list with hashes:

/tool.exe - **a1**

/main.c - **b1**



Commit message:

Improve main

Author:

Hector Alfaro

File list with hashes:

/tool.exe - **a1**

/main.c - **b2**

Parent commit hash:

$\epsilon_1 d_1$



Commit message:

Delete binary

Author:

Lars Schneider

File list with hashes:

/main.c - **b2**

Parent commit hash:

$\epsilon_2 d_2$

blobs



commits



Commit message:
My first commit

Author:
Lras Schneider

File list with hashes:
/tool.exe - **a1**
/main.c - **b1**

Commit message:
Improve main

Author:
Hector Alfaro

File list with hashes:
/tool.exe - **a1**
/main.c - **b2**

Parent commit hash:
e1 d1

Commit message:
Delete binary

Author:
Lars Schneider

File list with hashes:
/main.c - **b2**

Parent commit hash:
e2 d2

blobs





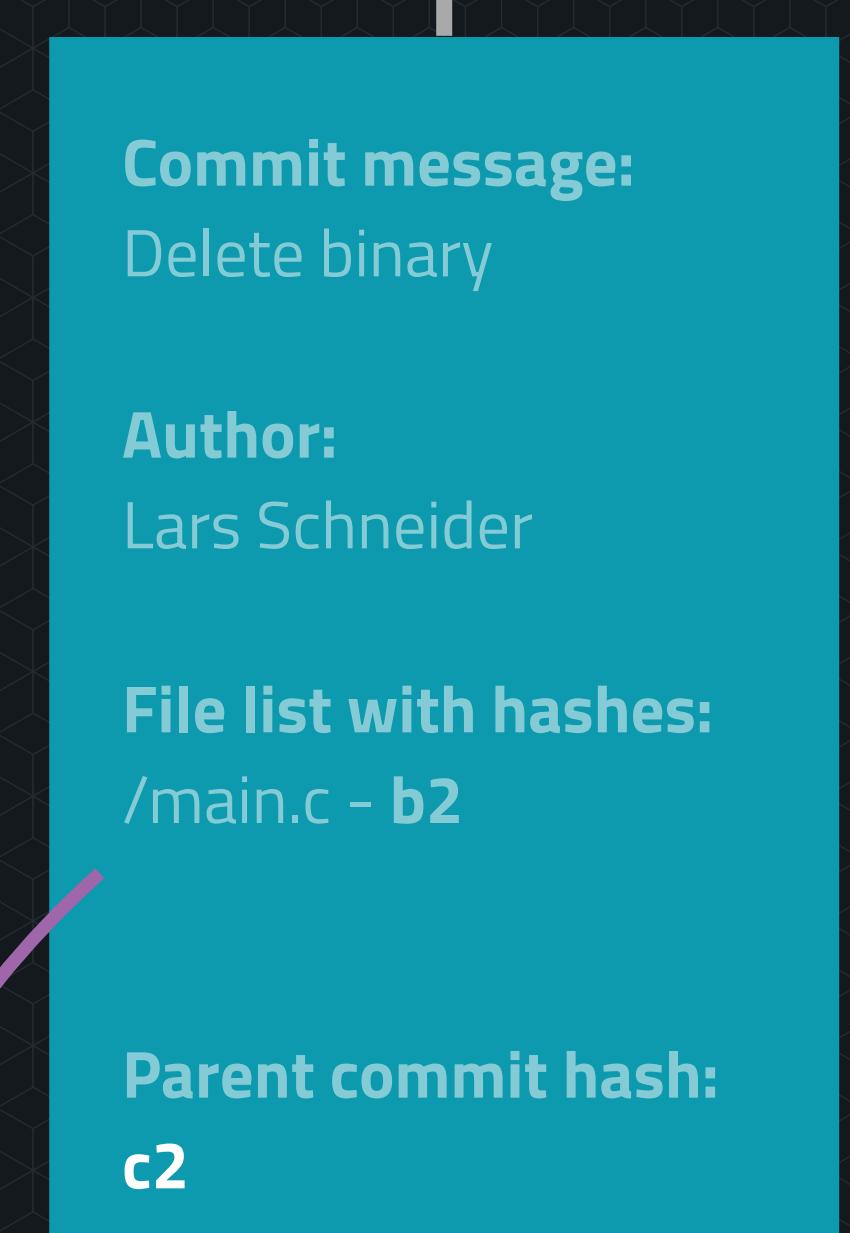
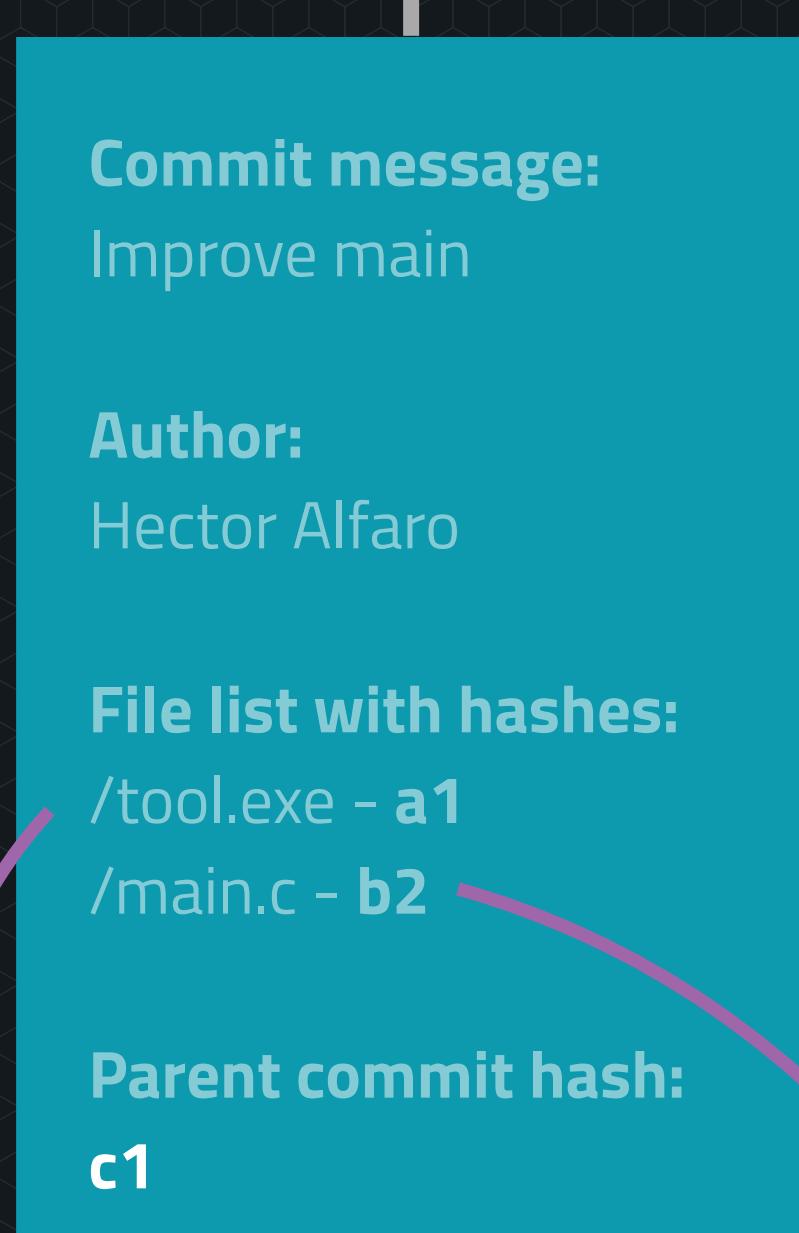
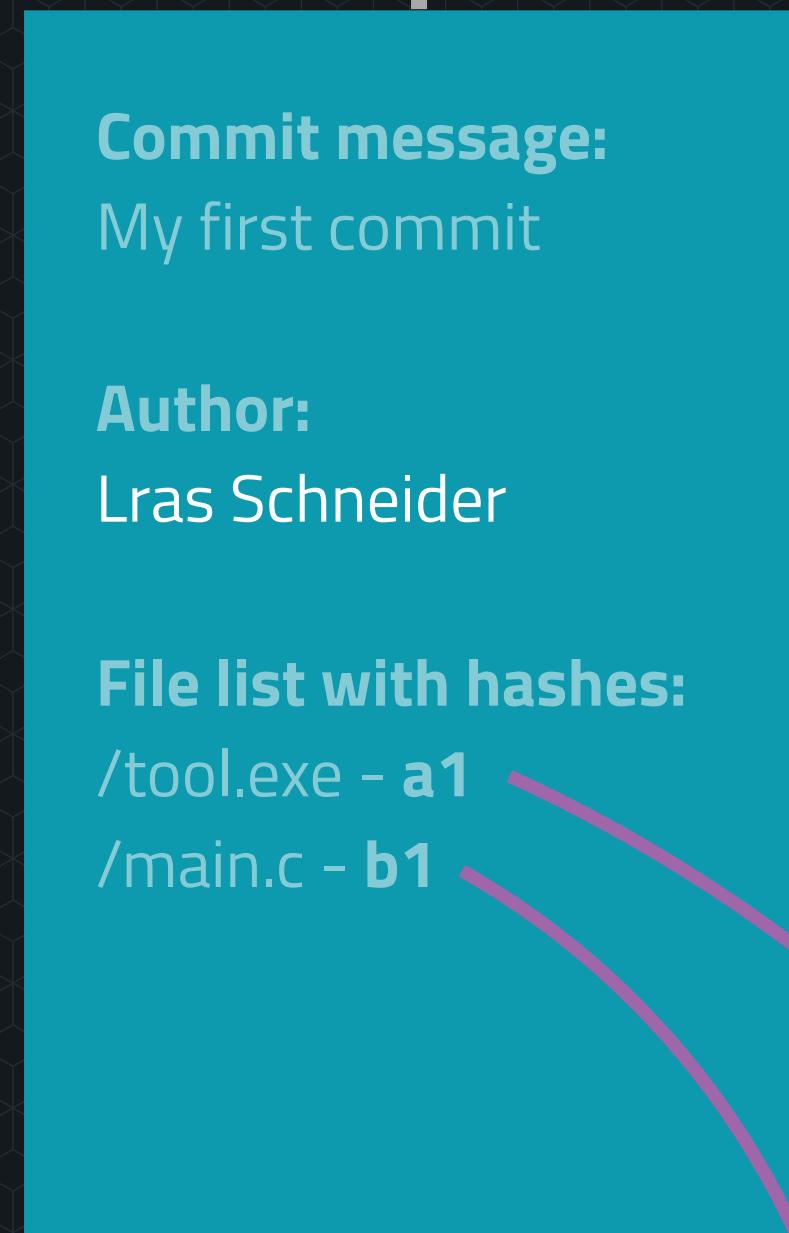
WHAT HAPPENS...
...IF YOU CHANGE AN AUTHOR?

commits

c1

c2

c3



Action

blobs

b1

a1

b2

commits

$\epsilon_1 e_1$

$\epsilon_2 e_2$

$\epsilon_3 e_3$

Commit message:
My first commit

Author:
Lars Schneider

File list with hashes:
`/tool.exe - a1`
`/main.c - b1`

Commit message:
Improve main

Author:
Hector Alfaro

File list with hashes:
`/tool.exe - a1`
`/main.c - b2`

Parent commit hash:
 $\epsilon_1 e_1$

Commit message:
Delete binary

Author:
Lars Schneider

File list with hashes:
`/main.c - b2`

Parent commit hash:
 $\epsilon_2 e_2$

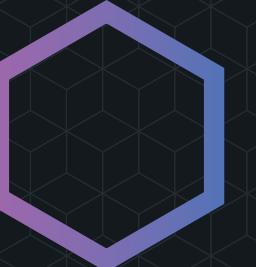
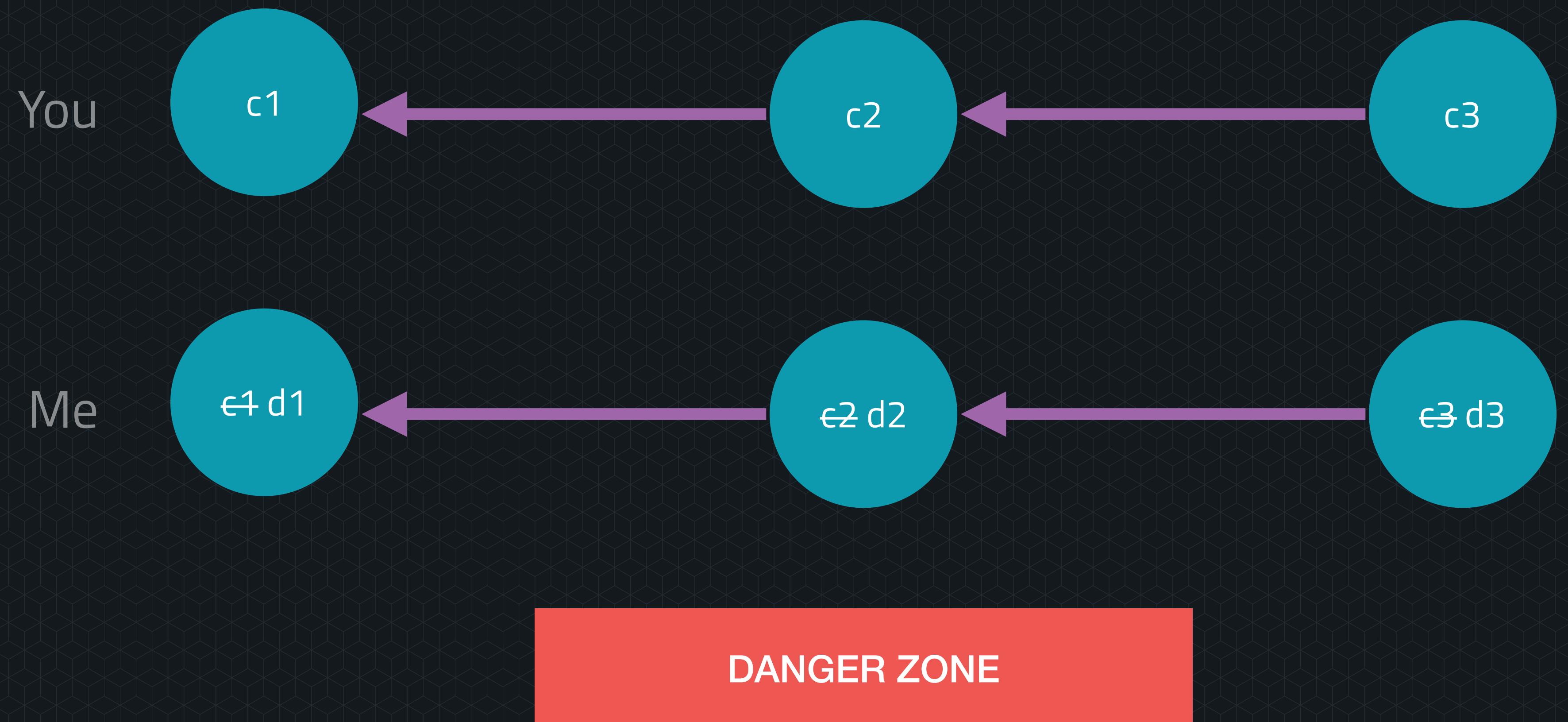
blobs

b1

a1

b2

WHAT IS THE PROBLEM?





WHAT HAPPENS IN GIT
STAYS IN GIT



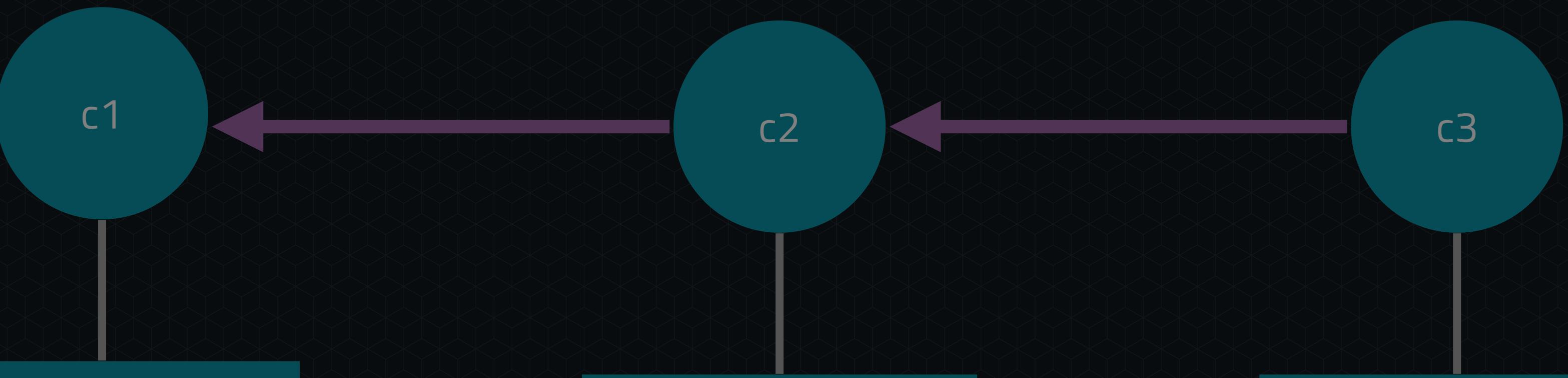
PRACTICE, FINALLY!



EXERCISE 1

FIND LARGE FILES

commits



Commit message:

My first commit

Author:

Lras Schneider

File list with hashes:

/tool.exe - **a1**

/main.c - **b1**

Commit message:

Improve main

Author:

Hector Alfaro

File list with hashes:

/tool.exe - **a1**

/main.c - **b2**

Parent commit hash:

c1

Commit message:

Delete binary

Author:

Lars Schneider

File list with hashes:

/main.c - **b2**

Parent commit hash:

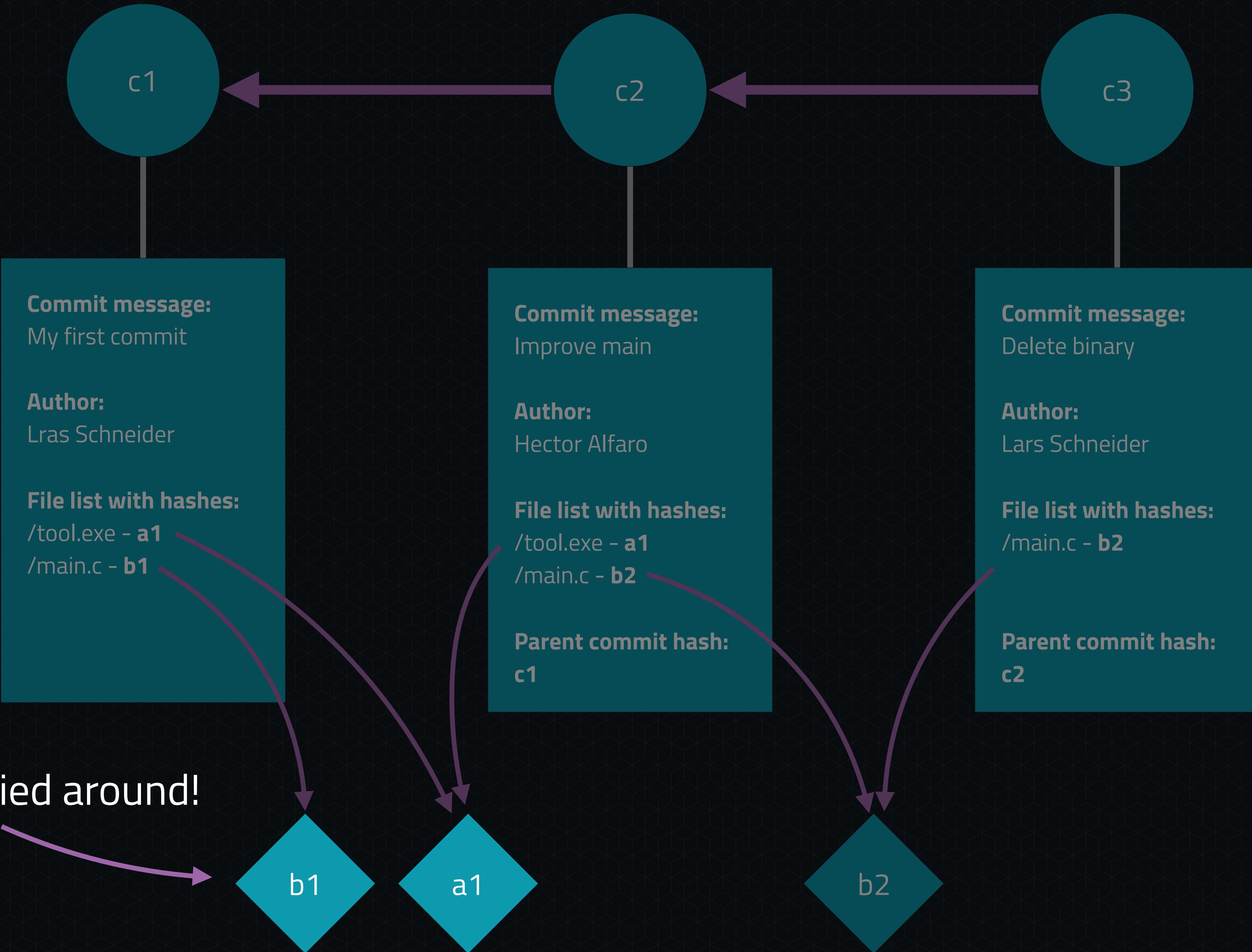
c2

blobs

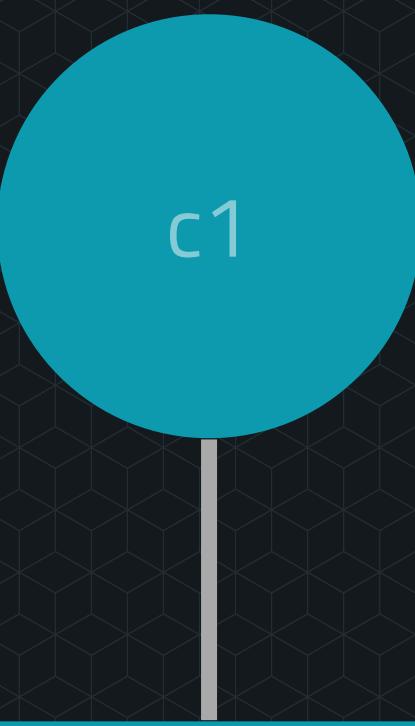


The only "visible" content!

commits



commits



Commit message:

My first commit

Author:

Lras Schneider

File list with hashes:

/tool.exe - **a1**

/main.c - **b1**



Commit message:

Improve main

Author:

Hector Alfaro

File list with hashes:

/tool.exe - **a1**

/main.c - **b2**

Parent commit hash:

c1



Commit message:

Delete binary

Author:

Lars Schneider

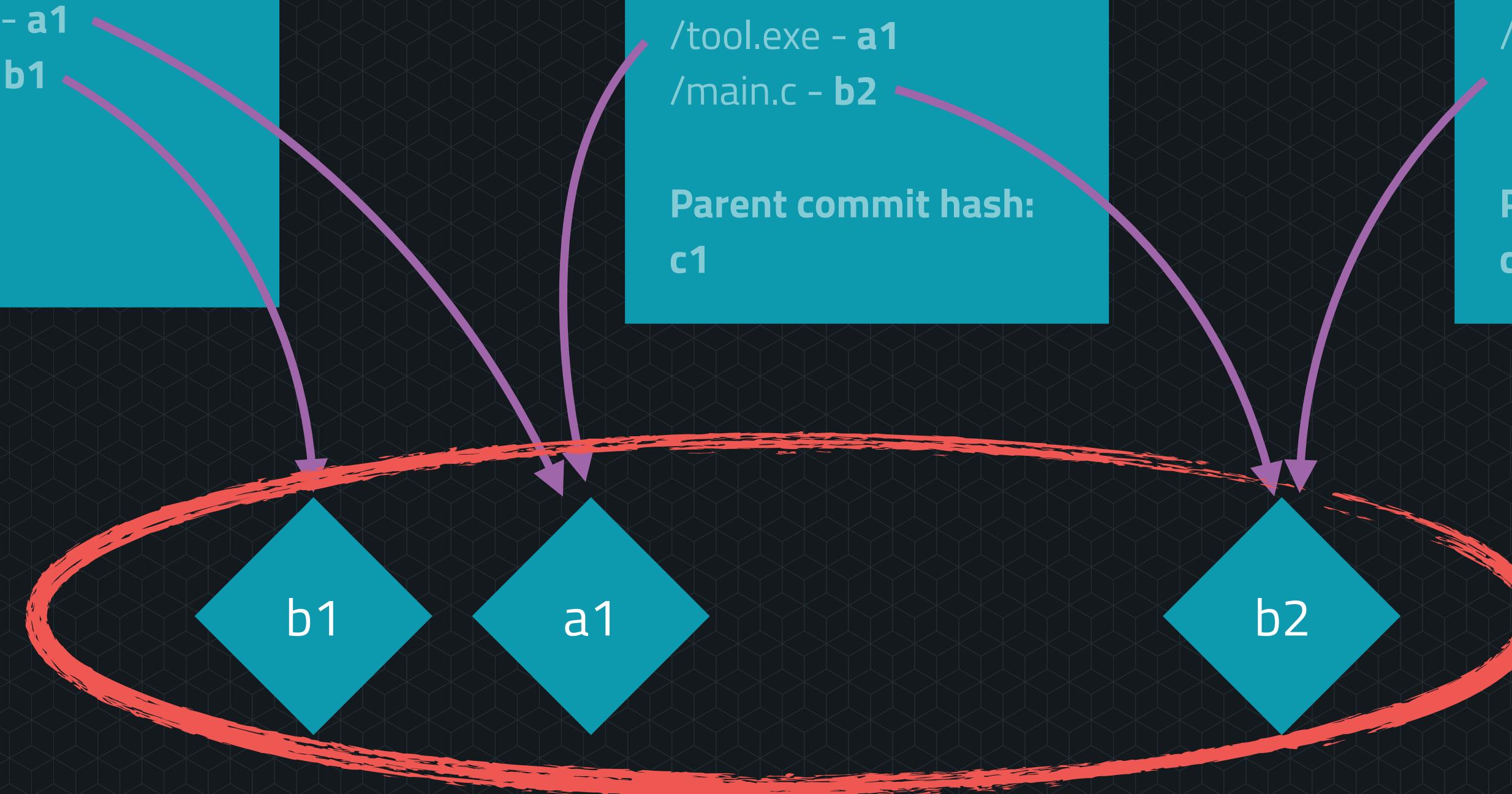
File list with hashes:

/main.c - **b2**

Parent commit hash:

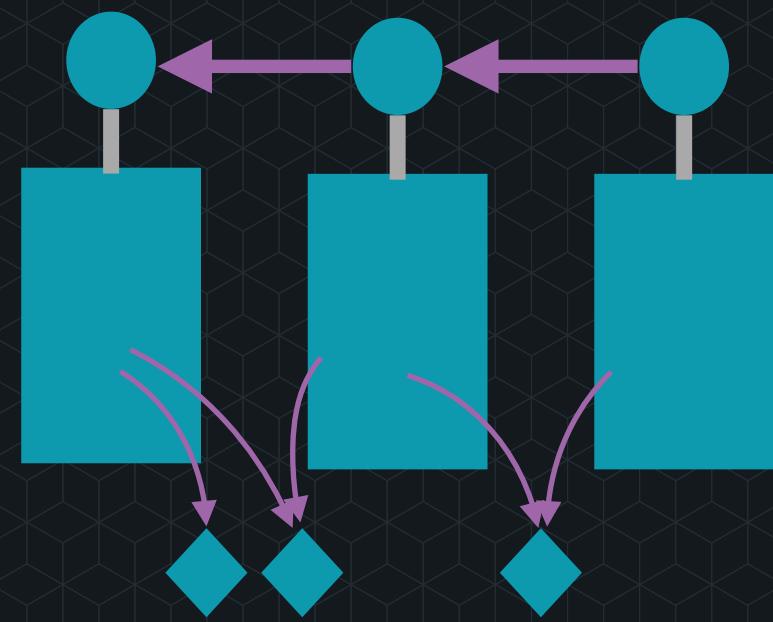
c2

blobs



Fork & Clone

<https://github.com/universeworkshops/deep-clean-exercise1>



```
$ git clone git@github.com:$YOUR_USER/deep-clean-exercise1.git  
  
Cloning into 'deep-clean-exercise1'...  
remote: Enumerating objects: 9, done.  
remote: Counting objects: 100% (9/9), done.  
remote: Compressing objects: 100% (7/7), done.  
remote: Total 9 (delta 1), reused 9 (delta 1), pack-reused 0  
Receiving objects: 100% (9/9), 1001.13 KiB | 264.00 KiB/s, done.  
Resolving deltas: 100% (1/1), done.
```

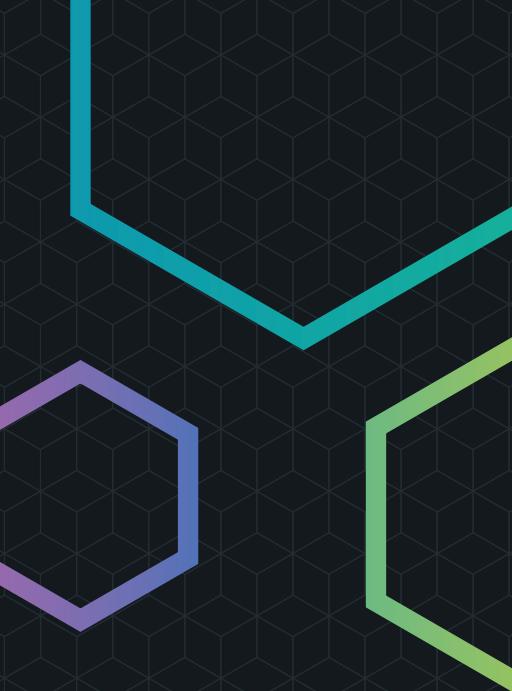
<https://github.com/universeworkshops/deep-clean-exercise1>

```
$ git clone git@github.com:$YOUR_USER/deep-clean-exercise1.git  
  
Cloning into 'deep-clean-exercise1'...  
remote: Enumerating objects: 9, done.  
remote: Counting objects: 100% (9/9), done.  
remote: Compressing objects: 100% (7/7), done.  
remote: Total 9 (delta 1), reused 9 (delta 1), pack-reused 0  
Receiving objects: 100% (9/9), 1001.13 KiB | 264.00 KiB/s, done.  
Resolving deltas: 100% (1/1), done.  
  
$ cd deep-clean-exercise1
```

```
$ git clone git@github.com:$YOUR_USER/deep-clean-exercise1.git  
  
Cloning into 'deep-clean-exercise1'...  
remote: Enumerating objects: 9, done.  
remote: Counting objects: 100% (9/9), done.  
remote: Compressing objects: 100% (7/7), done.  
remote: Total 9 (delta 1), reused 9 (delta 1), pack-reused 0  
Receiving objects: 100% (9/9), 1001.13 KiB | 264.00 KiB/s, done.  
Resolving deltas: 100% (1/1), done.  
  
$ cd deep-clean-exercise1  
  
$ du -hs .git/objects  
1.0M .git/objects
```



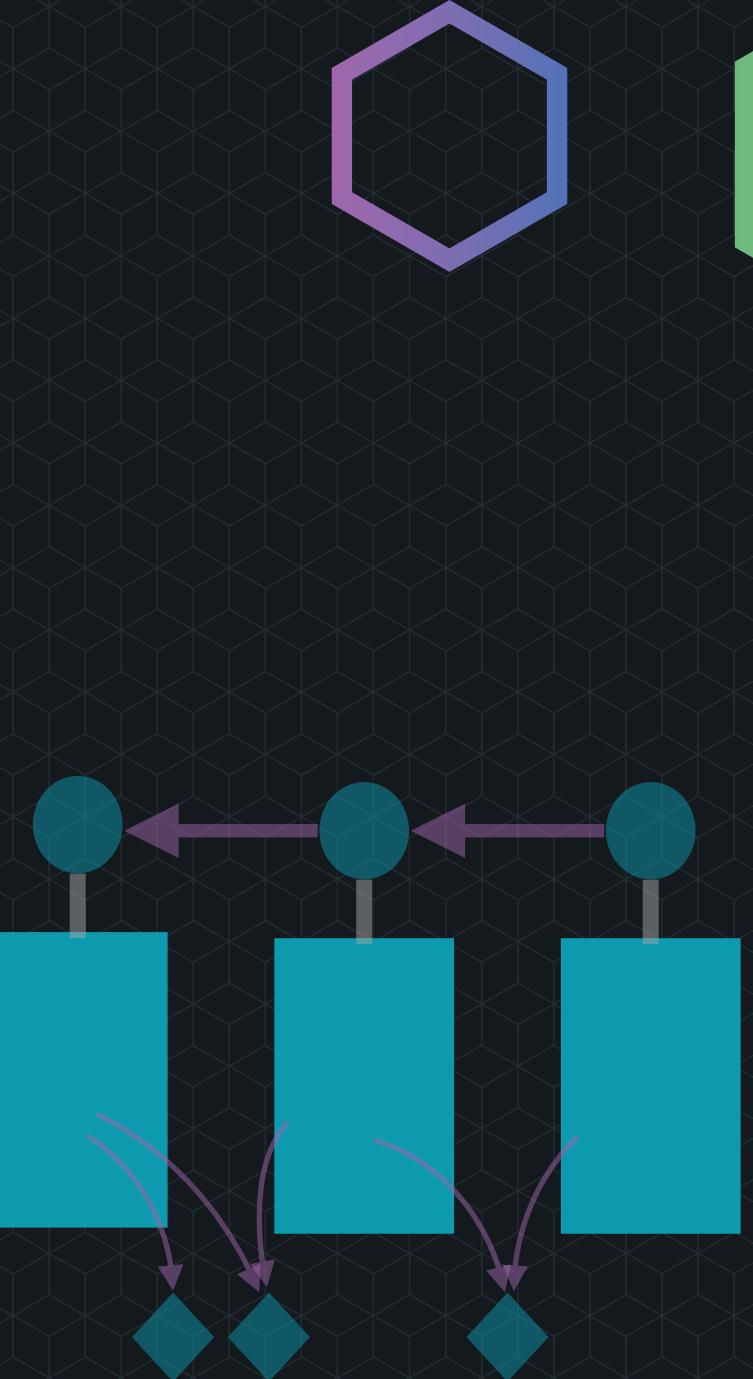
LOW LEVEL GIT COMMANDS AHEAD



```
$ git cat-file --batch-all-objects --batch-check  
03f04d1a50eeb4c831e925b9de9983e3547b496e blob 106  
12b95dbeb2e6a8a75fdb12ce9c86707e5658f1f3 tree 70  
130e75bcec86843476a7caadf460add6820ee360 tree 70  
7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b commit 239  
b26d812218ce88965b56fa5a5112fc67f941283b blob 1024000  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa blob 96  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec tree 34  
e73f2010475f2bec4a483d8d78cc49da0d678353 commit 190  
fddc5e0867190869523329b0a3690e6d4bc85358 commit 246
```

```
$ git cat-file --batch-all-objects --batch-check  
03f04d1a50eeb4c831e925b9de9983e3547b496e blob 106  
12b95dbeb2e6a8a75fdb12ce9c86707e5658f1f3 tree 70  
130e75bcec86843476a7caadf460add6820ee360 tree 70  
7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b commit 239  
b26d812218ce88965b56fa5a5112fc67f941283b blob 1024000  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa blob 96  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec tree 34  
e73f2010475f2bec4a483d8d78cc49da0d678353 commit 190  
fddc5e0867190869523329b0a3690e6d4bc85358 commit 246
```

```
$ git cat-file --batch-all-objects --batch-check  
03f04d1a50eeb4c831e925b9de9983e3547b496e blob 106  
12b95dbeb2e6a8a75fdb12ce9c86707e5658f1f3 tree 70  
130e75bcec86843476a7caadf460add6820ee360 tree 70  
7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b commit 239  
b26d812218ce88965b56fa5a5112fc67f941283b blob 1024000  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa blob 96  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec tree 34  
e73f2010475f2bec4a483d8d78cc49da0d678353 commit 190  
fddc5e0867190869523329b0a3690e6d4bc85358 commit 246
```



```
$ git cat-file --batch-all-objects --batch-check  
03f04d1a50eeb4c831e925b9de9983e3547b496e blob 106  
12b95dbeb2e6a8a75fdb12ce9c86707e5658f1f3 tree 70  
130e75bcec86843476a7caadf460add6820ee360 tree 70  
7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b commit 239  
b26d812218ce88965b56fa5a5112fc67f941283b blob 1024000  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa blob 96  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec tree 34  
e73f2010475f2bec4a483d8d78cc49da0d678353 commit 190  
fddc5e0867190869523329b0a3690e6d4bc85358 commit 246
```

```
$ git cat-file --batch-all-objects --batch-check  
03f04d1a50eeb4c831e925b9de9983e3547b496e blob 106  
12b95dbeb2e6a8a75fdb12ce9c86707e5658f1f3 tree 70  
130e75bcec86843476a7caadf460add6820ee360 tree 70  
7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b commit 239  
b26d812218ce88965b56fa5a5112fc67f941283b blob 1024000  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa blob 96  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec tree 34  
e73f2010475f2bec4a483d8d78cc49da0d678353 commit 190  
fddc5e0867190869523329b0a3690e6d4bc85358 commit 246
```

```
$ git cat-file --batch-all-objects --batch-check  
03f04d1a50eeb4c831e925b9de9983e3547b496e blob 106  
12b95dbeb2e6a8a75fdb12ce9c86707e5658f1f3 tree 70  
130e75bcec86843476a7caadf460add6820ee360 tree 70  
7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b commit 239  
b26d812218ce88965b56fa5a5112fc67f941283b blob 1024000  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa blob 96  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec tree 34  
e73f2010475f2bec4a483d8d78cc49da0d678353 commit 190  
fddc5e0867190869523329b0a3690e6d4bc85358 commit 246  
  
$ git rev-list --all --objects  
fddc5e0867190869523329b0a3690e6d4bc85358  
7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b  
e73f2010475f2bec4a483d8d78cc49da0d678353  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec  
03f04d1a50eeb4c831e925b9de9983e3547b496e main.c  
12b95dbeb2e6a8a75fdb12ce9c86707e5658f1f3  
b26d812218ce88965b56fa5a5112fc67f941283b tool.exe  
130e75bcec86843476a7caadf460add6820ee360  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa main.c
```

```
$ git cat-file --batch-all-objects --batch-check  
03f04d1a50eeb4c831e925b9de9983e3547b496e blob 106  
12b95dbeb2e6a8a75fdb12ce9c86707e5658f1f3 tree 70  
130e75bcec86843476a7caadf460add6820ee360 tree 70  
7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b commit 239  
b26d812218ce88965b56fa5a5112fc67f941283b blob 1024000  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa blob 96  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec tree 34  
e73f2010475f2bec4a483d8d78cc49da0d678353 commit 190  
fddc5e0867190869523329b0a3690e6d4bc85358 commit 246  
  
$ git rev-list --all --objects  
fddc5e0867190869523329b0a3690e6d4bc85358  
7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b  
e73f2010475f2bec4a483d8d78cc49da0d678353  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec  
03f04d1a50eeb4c831e925b9de9983e3547b496e main.c  
12b95dbeb2e6a8a75fdb12ce9c86707e5658f1f3  
b26d812218ce88965b56fa5a5112fc67f941283b tool.exe  
130e75bcec86843476a7caadf460add6820ee360  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa main.c
```

```
$ git cat-file --batch-all-objects --batch-check  
03f04d1a50eeb4c831e925b9de9983e3547b496e blob 106  
12b95dbeb2e6a8a75fdb12ce9c86707e5658f1f3 tree 70  
130e75bcec86843476a7caadf460add6820ee360 tree 70  
7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b commit 239  
b26d812218ce88965b56fa5a5112fc67f941283b blob 1024000  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa blob 96  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec tree 34  
e73f2010475f2bec4a483d8d78cc49da0d678353 commit 190  
fddc5e0867190869523329b0a3690e6d4bc85358 commit 246  
  
$ git rev-list --all --objects  
fddc5e0867190869523329b0a3690e6d4bc85358  
7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b  
e73f2010475f2bec4a483d8d78cc49da0d678353  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec  
03f04d1a50eeb4c831e925b9de9983e3547b496e main.c  
12b95dbeb2e6a8a75fdb12ce9c86707e5658f1f3  
b26d812218ce88965b56fa5a5112fc67f941283b tool.exe  
130e75bcec86843476a7caadf460add6820ee360  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa main.c  
  
$ git-find-large-files  
1000 Deleted tool.exe
```

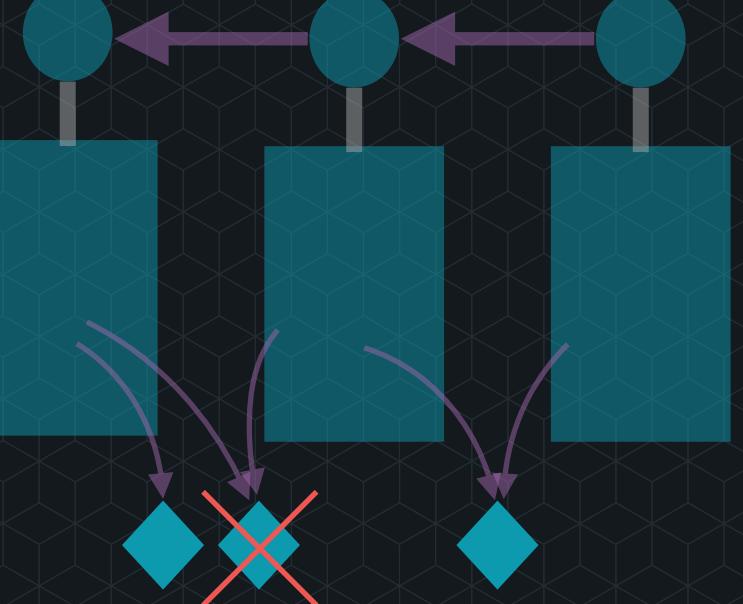
Action: <https://github.com/larsxschneider/git-repo-analysis>



EXERCISE 2

PURGE FILES

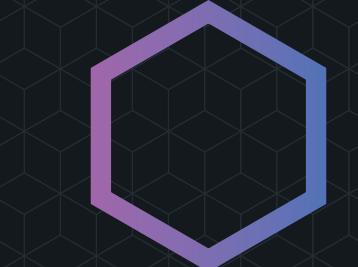
```
$ git filter-branch --index-filter 'git rm --cached --ignore-unmatch tool.exe' -- --all  
Rewrite e73f2010475f2bec4a483d8d78cc49da0d678353 (1/3) (0 seconds passed, remaining 0 predicted)  
rm 'tool.exe'  
Rewrite 7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b (2/3) (1 seconds passed, remaining 0 predicted)  
rm 'tool.exe'  
Rewrite fddc5e0867190869523329b0a3690e6d4bc85358 (2/3) (1 seconds passed, remaining 0 predicted)  
Ref 'refs/heads/master' was rewritten  
Ref 'refs/remotes/origin/master' was rewritten  
WARNING: Ref 'refs/remotes/origin/master' is unchanged
```



```
$ git filter-branch --index-filter 'git rm --cached --ignore-unmatch tool.exe' -- --all  
Rewrite e73f2010475f2bec4a483d8d78cc49da0d678353 (1/3) (0 seconds passed, remaining 0 predicted)  
rm 'tool.exe'  
Rewrite 7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b (2/3) (1 seconds passed, remaining 0 predicted)  
rm 'tool.exe'  
Rewrite fddc5e0867190869523329b0a3690e6d4bc85358 (2/3) (1 seconds passed, remaining 0 predicted)  
Ref 'refs/heads/master' was rewritten  
Ref 'refs/remotes/origin/master' was rewritten  
WARNING: Ref 'refs/remotes/origin/master' is unchanged
```

```
$ git filter-branch --index-filter 'git rm --cached --ignore-unmatch tool.exe' -- --all  
Rewrite e73f2010475f2bec4a483d8d78cc49da0d678353 (1/3) (0 seconds passed, remaining 0 predicted)  
rm 'tool.exe'  
Rewrite 7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b (2/3) (1 seconds passed, remaining 0 predicted)  
rm 'tool.exe'  
Rewrite fddc5e0867190869523329b0a3690e6d4bc85358 (2/3) (1 seconds passed, remaining 0 predicted)  
Ref 'refs/heads/master' was rewritten  
Ref 'refs/remotes/origin/master' was rewritten  
WARNING: Ref 'refs/remotes/origin/master' is unchanged
```

```
$ git-purge-files '/tool\.exe$'
```



Action



100x faster!

```
$ git filter-branch --index-filter 'git rm --cached --ignore-unmatch tool.exe' -- --all  
Rewrite e73f2010475f2bec4a483d8d78cc49da0d678353 (1/3) (0 seconds passed, remaining 0 predicted)  
rm 'tool.exe'  
Rewrite 7c2b6b0c5faad8cd3f6231b6abbfa87b2882e93b (2/3) (1 seconds passed, remaining 0 predicted)  
rm 'tool.exe'  
Rewrite fddc5e0867190869523329b0a3690e6d4bc85358 (2/3) (1 seconds passed, remaining 0 predicted)  
Ref 'refs/heads/master' was rewritten  
Ref 'refs/remotes/origin/master' was rewritten  
WARNING: Ref 'refs/remotes/origin/master' is unchanged  
  
$ git-purge-files '/tool\.exe$'  
  
$ git push origin --mirror  
  
Enumerating objects: 13, done.  
Counting objects: 100% (13/13), done.  
Delta compression using up to 8 threads.  
Compressing objects: 100% (8/8), done.  
Writing objects: 100% (13/13), 1001.45 KiB | 9.19 MiB/s, done.  
Total 13 (delta 3), reused 13 (delta 3)  
remote: Resolving deltas: 100% (3/3), done.  
To github.com:$YOUR_USER/deep-clean-exercise1.git  
 + fddc5e0...37099ab master -> master (forced update)
```



NEW CLONE

```
$ git clone git@github.com:$YOUR_USER/deep-clean-exercise1.git
```

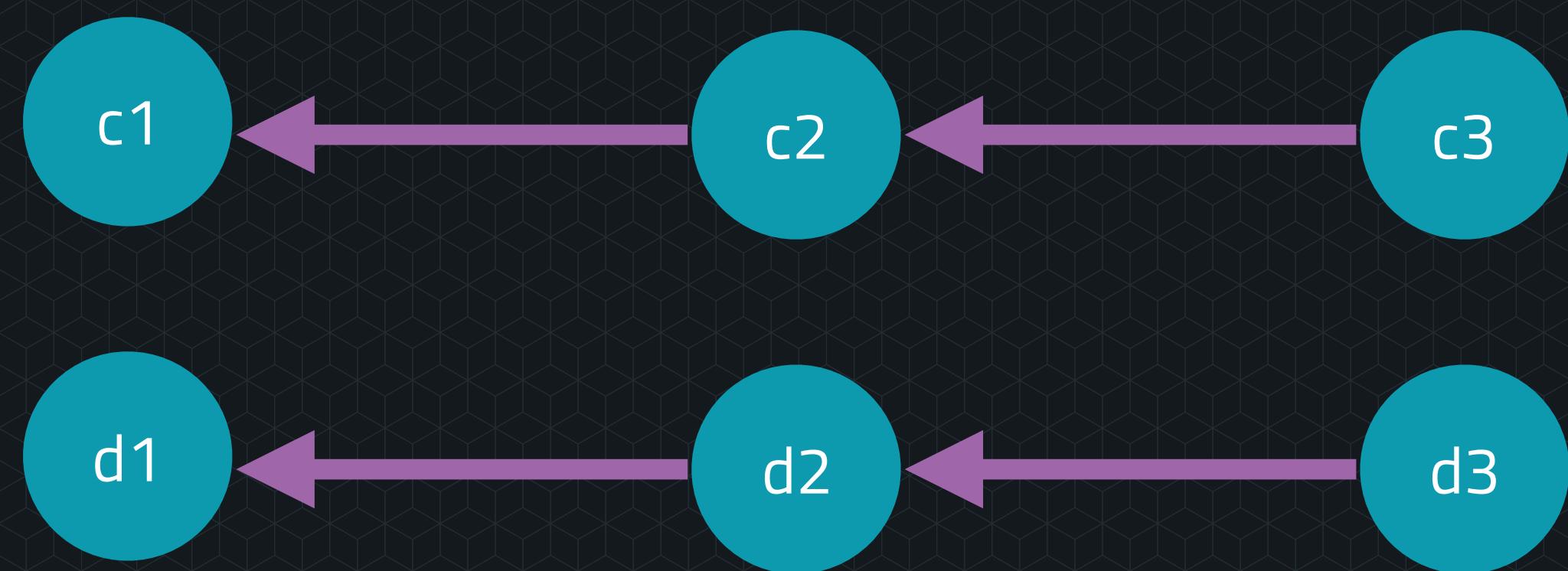
```
Cloning into 'deep-clean-exercise1'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 1), reused 7 (delta 1), pack-reused 0
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.
```

```
$ git clone git@github.com:$YOUR_USER/deep-clean-exercise1.git  
  
Cloning into 'deep-clean-exercise1'...  
remote: Enumerating objects: 7, done.  
remote: Counting objects: 100% (7/7), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 7 (delta 1), reused 7 (delta 1), pack-reused 0  
Receiving objects: 100% (7/7), done.  
Resolving deltas: 100% (1/1), done.  
  
$ git cat-file --batch-all-objects --batch-check  
  
03f04d1a50eeb4c831e925b9de9983e3547b496e blob 106  
37099abd9c929e3df632803435cf090cf981ad57 commit 246  
aa9312514ca5446e589bf7b98f31309b9a952548 commit 239  
ad48b0b205fda358bd4cc914d643bef6b582e1e commit 190  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa blob 96  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec tree 34  
f043ba7e74fabb92b08ace0f86a05b42a197f9bd tree 34
```

```
$ git clone git@github.com:$YOUR_USER/deep-clean-exercise1.git  
  
Cloning into 'deep-clean-exercise1'...  
remote: Enumerating objects: 7, done.  
remote: Counting objects: 100% (7/7), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 7 (delta 1), reused 7 (delta 1), pack-reused 0  
Receiving objects: 100% (7/7), done.  
Resolving deltas: 100% (1/1), done.  
  
$ git cat-file --batch-all-objects --batch-check  
  
03f04d1a50eeb4c831e925b9de9983e3547b496e blob 106  
37099abd9c929e3df632803435cf090cf981ad57 commit 246  
aa9312514ca5446e589bf7b98f31309b9a952548 commit 239  
ad48b0b205fda358bd4cc914d643bef6b582e1e commit 190  
b69a1bd969b4b6a581d62a0a43a0d3d71983f3fa blob 96  
b88ca904dac19a19e90ac0a0764d2cc6dc8414ec tree 34  
f043ba7e74fabb92b08ace0f86a05b42a197f9bd tree 34
```

IF YOU REWRITE HISTORY...

1. Colleagues: Push all in-progress work
2. Colleagues: Delete local repository
3. Colleagues: Stop working
4. You: **git clone --mirror <url>**
5. You: Rewrite the history
6. You: Disable branch protection, **git push --mirror <url>**, enable branch protection
7. Colleagues: Clone the repository



GIT-PURGE-FILES

```
$ git-purge-files 'regex-1' 'regex-2' ... 'regex-n'
```

Regular Expression

```
# Remove tool.exe in root directory  
$ git-purge-files '^/tool\.exe$'
```

Regular Expression

```
# Remove tool.exe in root directory  
$ git-purge-files '^/tool\.exe$'
```

Regular Expression

```
# Remove tool.exe in root directory  
$ git-purge-files '^/tool\.exe$'
```

Regular Expression

```
# Remove tool.exe in root directory  
$ git-purge-files '^/tool\.exe$'
```

Regular Expression

```
# Remove tool.exe in root directory  
$ git-purge-files '^/tool\.exe$'
```

```
# Remove tool.exe in all directories  
$ git-purge-files '/tool\.exe$'
```

/tool.exe
/foo/tool.exe
/some/other/tool.exe

Regular Expression

```
# Remove tool.exe in root directory  
$ git-purge-files '^/tool\.exe$'
```

```
# Remove tool.exe in all directories  
$ git-purge-files '/tool\.exe$'
```

```
# Remove all *.exe files  
$ git-purge-files '\.exe$'
```

/tool.exe
/hello.exe
/some/other/binary.exe

Regular Expression

```
# Remove tool.exe in root directory  
$ git-purge-files '^/tool\.exe$'
```

```
# Remove tool.exe in all directories  
$ git-purge-files '/tool\.exe$'
```

```
# Remove all *.exe files  
$ git-purge-files '\.exe$'
```

```
# Remove bin directory in root  
$ git-purge-files '^/bin/'
```

/bin/file.exe
/bin/other/lib.dll

Regular Expression

```
# Remove tool.exe in root directory  
$ git-purge-files '^/tool\.exe$'
```

```
# Remove tool.exe in all directories  
$ git-purge-files '/tool\.exe$'
```

```
# Remove all *.exe files  
$ git-purge-files '\.exe$'
```

```
# Remove bin directory in root  
$ git-purge-files '^/bin/'
```

```
# Remove all bin directories and all *.exe files  
$ git-purge-files '/bin/' '\.exe$'
```

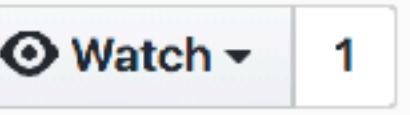
Regular Expression



EXERCISE 3

FIND AND PURGE LARGE DIRECTORIES

 [universeworkshops / deep-clean-exercise3](#) Private

 Watch 1  Star 0  Fork 0

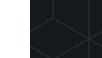
 [Code](#)  [Issues 0](#)  [Pull requests 0](#)  [Projects 0](#)  [Wiki](#)  [Insights](#)  [Settings](#)

No description, website, or topics provided. 

[Manage topics](#)

 4 commits  1 branch  0 releases  1 contributor

Branch: [master](#)  New pull request  Create new file  Upload files  Find file  Clone or download

 larsxschneider	add main.c	Latest commit <code>73dac62</code> 15 minutes ago
	Initial commit	20 minutes ago
	add main.c	5 minutes ago
		

deep-clean-exercise3

Fork & clone: <https://github.com/universeworkshops/deep-clean-exercise3>

```
$ git clone https://github.com/$YOUR_USER/deep-clean-exercise3.git  
  
Cloning into 'deep-clean-exercise3'...  
remote: Enumerating objects: 446, done.  
remote: Counting objects: 100% (446/446), done.  
remote: Compressing objects: 100% (407/407), done.  
remote: Total 446 (delta 34), reused 446 (delta 34), pack-reused 0  
Receiving objects: 100% (446/446), 574.24 KiB | 1.21 MiB/s, done.  
Resolving deltas: 100% (34/34), done.
```

Fork & clone: <https://github.com/universeworkshops/deep-clean-exercise3>

```
$ git clone https://github.com/$YOUR_USER/deep-clean-exercise3.git  
  
Cloning into 'deep-clean-exercise3'...  
remote: Enumerating objects: 446, done.  
remote: Counting objects: 100% (446/446), done.  
remote: Compressing objects: 100% (407/407), done.  
remote: Total 446 (delta 34), reused 446 (delta 34), pack-reused 0  
Receiving objects: 100% (446/446), 574.24 KiB | 1.21 MiB/s, done.  
Resolving deltas: 100% (34/34), done.  
  
$ git-find-large-files
```

<https://github.com/universeworkshops/deep-clean-exercise3>

```
$ git clone https://github.com/$YOUR_USER/deep-clean-exercise3.git  
Cloning into 'deep-clean-exercise3'...  
remote: Enumerating objects: 446, done.  
remote: Counting objects: 100% (446/446), done.  
remote: Compressing objects: 100% (407/407), done.  
remote: Total 446 (delta 34), reused 446 (delta 34), pack-reused 0  
Receiving objects: 100% (446/446), 574.24 KiB | 1.21 MiB/s, done.  
Resolving deltas: 100% (34/34), done.  
  
$ git-find-large-files  
# NO RESULT !
```

```
$ git clone https://github.com/$YOUR_USER/deep-clean-exercise3.git  
Cloning into 'deep-clean-exercise3'...  
remote: Enumerating objects: 446, done.  
remote: Counting objects: 100% (446/446), done.  
remote: Compressing objects: 100% (407/407), done.  
remote: Total 446 (delta 34), reused 446 (delta 34), pack-reused 0  
Receiving objects: 100% (446/446), 574.24 KiB | 1.21 MiB/s, done.  
Resolving deltas: 100% (34/34), done.  
  
$ git-find-large-files  
# NO RESULT !  
  
$ git-find-dirs-deleted-files
```

```
$ git clone https://github.com/$YOUR_USER/deep-clean-exercise3.git  
Cloning into 'deep-clean-exercise3'...  
remote: Enumerating objects: 446, done.  
remote: Counting objects: 100% (446/446), done.  
remote: Compressing objects: 100% (407/407), done.  
remote: Total 446 (delta 34), reused 446 (delta 34), pack-reused 0  
Receiving objects: 100% (446/446), 574.24 KiB | 1.21 MiB/s, done.  
Resolving deltas: 100% (34/34), done.  
  
$ git-find-large-files  
# NO RESULT !  
  
$ git-find-dirs-deleted-files  
395 deleted 3rdparty
```

```
$ git clone https://github.com/$YOUR_USER/deep-clean-exercise3.git  
Cloning into 'deep-clean-exercise3'...  
remote: Enumerating objects: 446, done.  
remote: Counting objects: 100% (446/446), done.  
remote: Compressing objects: 100% (407/407), done.  
remote: Total 446 (delta 34), reused 446 (delta 34), pack-reused 0  
Receiving objects: 100% (446/446), 574.24 KiB | 1.21 MiB/s, done.  
Resolving deltas: 100% (34/34), done.  
  
$ git-find-large-files  
# NO RESULT !  
  
$ git-find-dirs-deleted-files  
395 deleted 3rdparty  
  
$ git-purge-files '^/3rdparty/'
```

```
$ git clone https://github.com/$YOUR_USER/deep-clean-exercise3.git  
  
Cloning into 'deep-clean-exercise3'...  
remote: Enumerating objects: 446, done.  
remote: Counting objects: 100% (446/446), done.  
remote: Compressing objects: 100% (407/407), done.  
remote: Total 446 (delta 34), reused 446 (delta 34), pack-reused 0  
Receiving objects: 100% (446/446), 574.24 KiB | 1.21 MiB/s, done.  
Resolving deltas: 100% (34/34), done.  
  
$ git-find-large-files  
  
# NO RESULT !  
  
$ git-find-dirs-deleted-files  
  
395 deleted 3rdparty  
  
$ git-purge-files '^/3rdparty/'  
  
$ git-find-dirs-deleted-files  
  
# NO RESULT !
```



EXERCISE 4

FIX COMMIT AUTHOR ATTRIBUTION

 [universeworkshops / deep-clean-exercise4](#) Private

[Watch](#) 1 [Star](#) 0 [Fork](#) 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#) [Settings](#)

Branch: master ▾

Commits on Sep 22, 2018

- Delete binary**
 larsxschneider committed a day ago [Edit](#) [37099ab](#) [Compare](#)
- Improve main**
 hectorsector committed a day ago [Edit](#) [aa93125](#) [Compare](#)
- My first commit**
 Lras Schneider committed a day ago [Edit](#) [ad48b0b](#) [Compare](#)



Fork & clone: <https://github.com/universeworkshops/deep-clean-exercise4>

```
$ git log
```

```
commit 37099abd9c929e3df632803435cf090cf981ad57 (HEAD -> master, origin/master, origin/HEAD)
Author: Lars Schneider <larsxschneider@github.com>

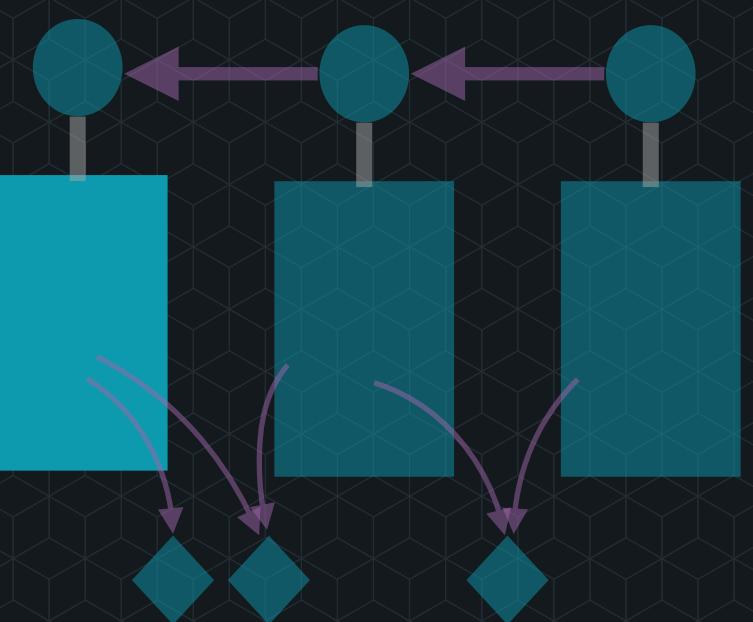
    Delete binary

commit aa9312514ca5446e589bf7b98f31309b9a952548
Author: Hector Alfaro <hectorsector@github.com>

    Improve main

commit ad48b0b205fda358bd4cc914d643bef6b582e1e
Author: Lras Schneider <lras@wrong-email.com>

    My first commit
```



```
$ git filter-branch --env-filter '  
OLD_EMAIL="lras@wrong-email.com"  
CORRECT_NAME="Lars Schneider"  
CORRECT_EMAIL="larsxschneider@github.com"  
  
if [ "$GIT_AUTHOR_EMAIL" = "$OLD_EMAIL" ]  
then  
    export GIT_AUTHOR_NAME="$CORRECT_NAME"  
    export GIT_AUTHOR_EMAIL="$CORRECT_EMAIL"  
fi  
,
```

```
$ git filter-branch --env-filter '
  OLD_EMAIL="lras@wrong-email.com"
  CORRECT_NAME="Lars Schneider"
  CORRECT_EMAIL="larsxschneider@github.com"

  if [ "$GIT_AUTHOR_EMAIL" = "$OLD_EMAIL" ]
  then
    export GIT_AUTHOR_NAME="$CORRECT_NAME"
    export GIT_AUTHOR_EMAIL="$CORRECT_EMAIL"
  fi
'

$ git-change-author 'lras@wrong-email.com' 'Lars Schneider' 'larsxschneider@github.com'

Rewrite 37099abd9c929e3df632803435cf090cf981ad57 (3/3) (0 seconds passed, remaining 0 predicted)
Ref 'refs/heads/master' was rewritten
Ref 'refs/remotes/origin/master' was rewritten
WARNING: Ref 'refs/remotes/origin/master' is unchanged
```

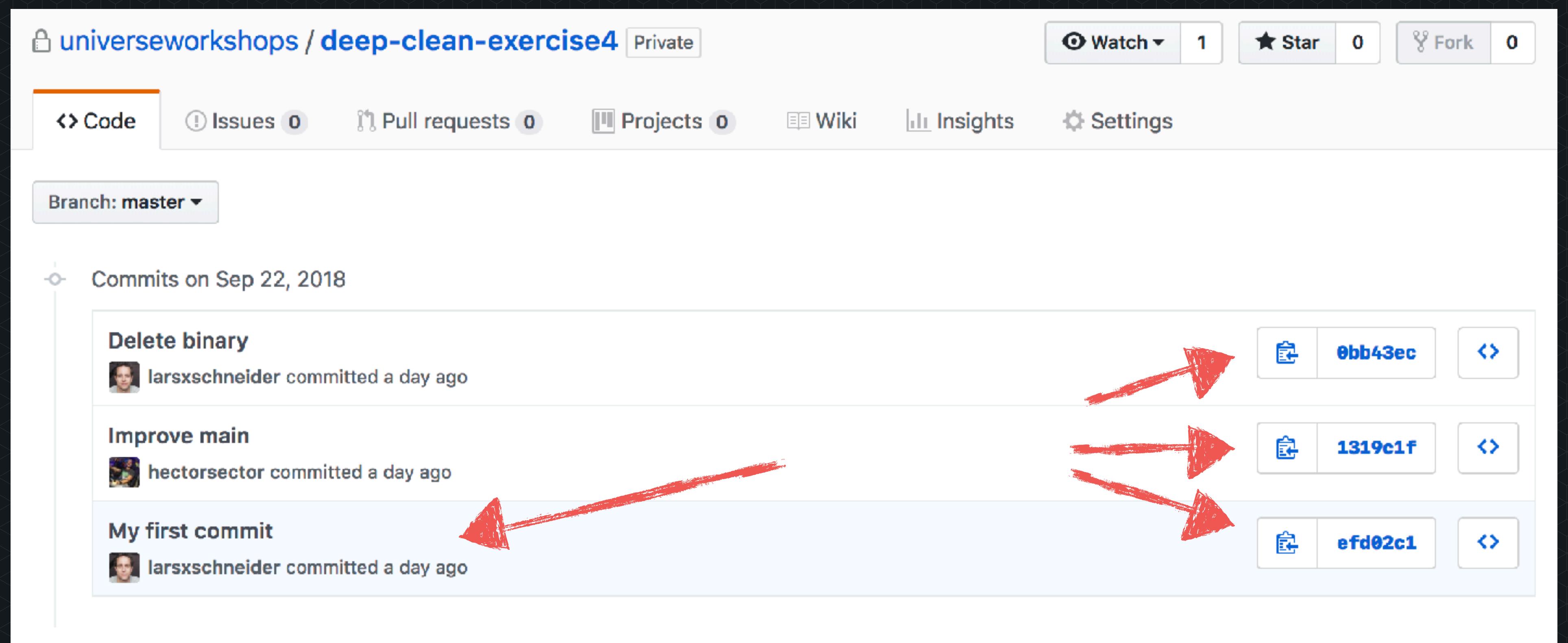
```
$ git filter-branch --env-filter '
  OLD_EMAIL="lras@wrong-email.com"
  CORRECT_NAME="Lars Schneider"
  CORRECT_EMAIL="larsxschneider@github.com"

  if [ "$GIT_AUTHOR_EMAIL" = "$OLD_EMAIL" ]
  then
    export GIT_AUTHOR_NAME="$CORRECT_NAME"
    export GIT_AUTHOR_EMAIL="$CORRECT_EMAIL"
  fi
'

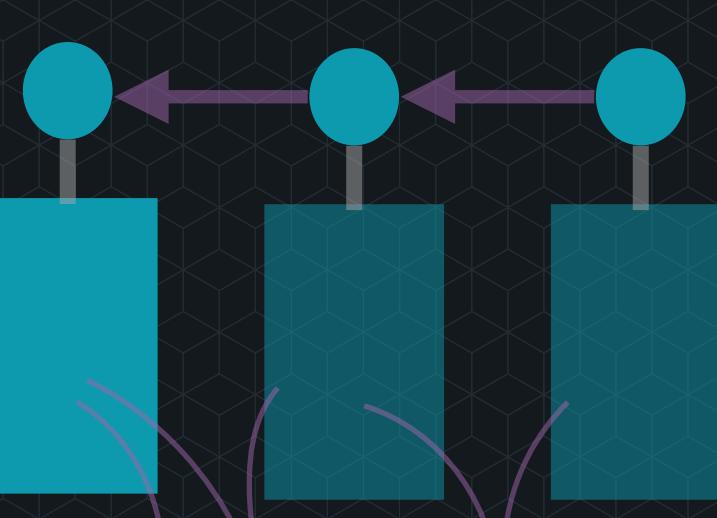
$ git-change-author 'lras@wrong-email.com' 'Lars Schneider' 'larsxschneider@github.com'

Rewrite 37099abd9c929e3df632803435cf090cf981ad57 (3/3) (0 seconds passed, remaining 0 predicted)
Ref 'refs/heads/master' was rewritten
Ref 'refs/remotes/origin/master' was rewritten
WARNING: Ref 'refs/remotes/origin/master' is unchanged

$ git push origin --mirror
```

A screenshot of a GitHub repository page. The repository name is "universeworkshops / deep-clean-exercise4" and it is marked as "Private". The "Code" tab is selected. There are 1 watch, 0 stars, and 0 forks. The branch is set to "master". A dropdown shows "Commits on Sep 22, 2018". Three commits are listed:

- Delete binary** by larsxschneider committed a day ago
- Improve main** by hectorsector committed a day ago
- My first commit** by larsxschneider committed a day ago

Red arrows point from the "My first commit" commit to the "Delete binary" and "Improve main" commits. To the right of the commits are three cards with commit details: `0bb43ec`, `1319c1f`, and `efd02c1`.



EXERCISE 5

HOMEWORK



GIT REPO ANALYSIS TOOLS

<https://github.com/larsxschneider/git-repo-analysis>



- git-find-large-files ✓
- git-find-dirs-deleted-files ✓
- git-purge-files ✓
- git-change-author ✓
- git-find-dirs-many-files
- git-find-dirs-unwanted
- git-find-ignored-files
- git-find-lfs-extensions
- git-normalize-pathnames
- ...

GIT SIZER

<https://github.com/github/git-sizer>

```
$ git-sizer --verbose
Processing blobs: 1652370
Processing trees: 3396199
Processing commits: 722647
Matching commits to trees: 722647
Processing annotated tags: 534
Processing references: 539
```

Name	Value	Level of concern
Overall repository size		
* Commits		
* Count	723 k	*
* Total size	525 MiB	**
* Trees		
* Count	3.40 M	**
* Total size	9.00 GiB	****
* Total tree entries	264 M	*****
* Blobs		
* Count	1.65 M	*
* Total size	55.8 GiB	*****
* Annotated tags		
* Count	534	
* References		

WRAP UP - WHAT DID WE LEARN?

- We learned how Git works
- We learned how to find large files/directories in Git repos
- We learned how to entirely purge files/directories from Git repos
- We learned how to fix commit author contribution



WRAP UP - WHAT DID WE LEARN?

- We learned how Git works
- We learned how to find large files/directories in Git repos
- We learned how to entirely purge files/directories from Git repos
- We learned how to fix commit author contribution

Questions?





U N I V E R S E