Git On My Level

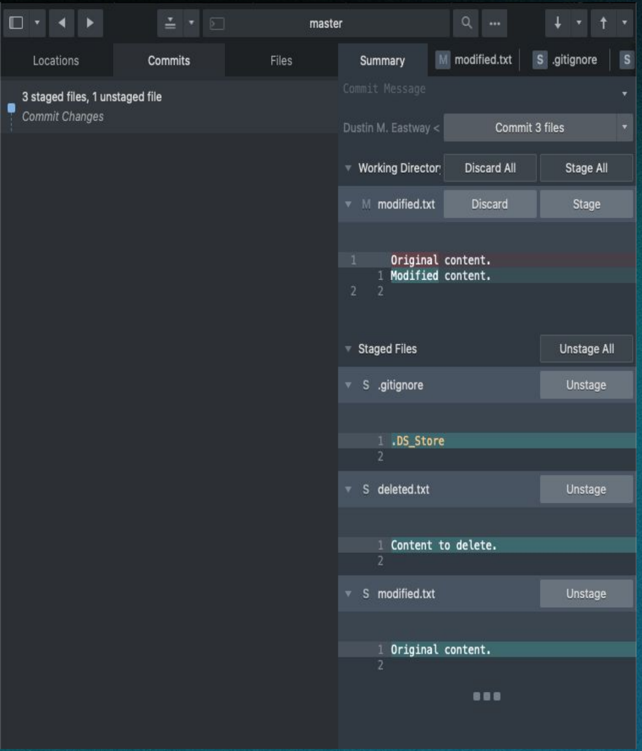
Bridging Knowledge with Practical Solutions

https://github.com/DustinMEastway/git-on-my-level

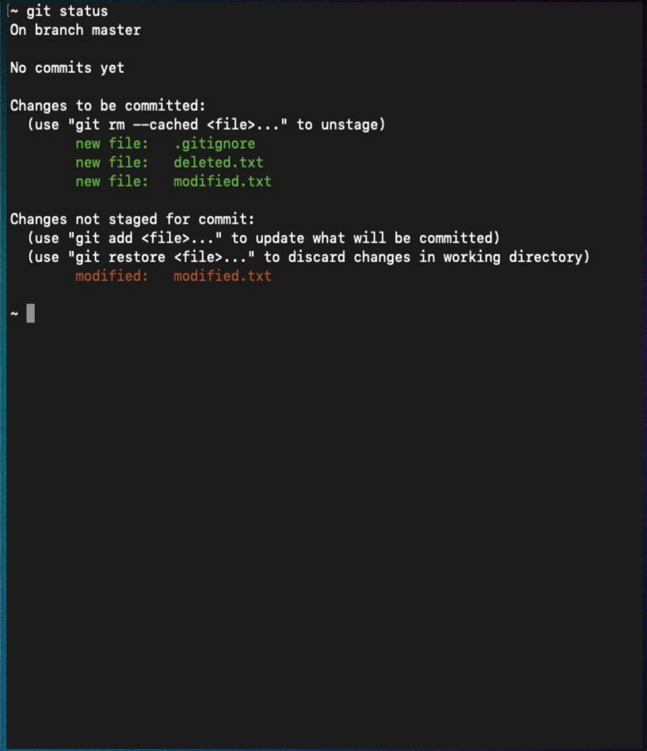
Dustin M. Eastway
CodeMash
2025-01-17

Term: Client

The tool you use to interact with Git.

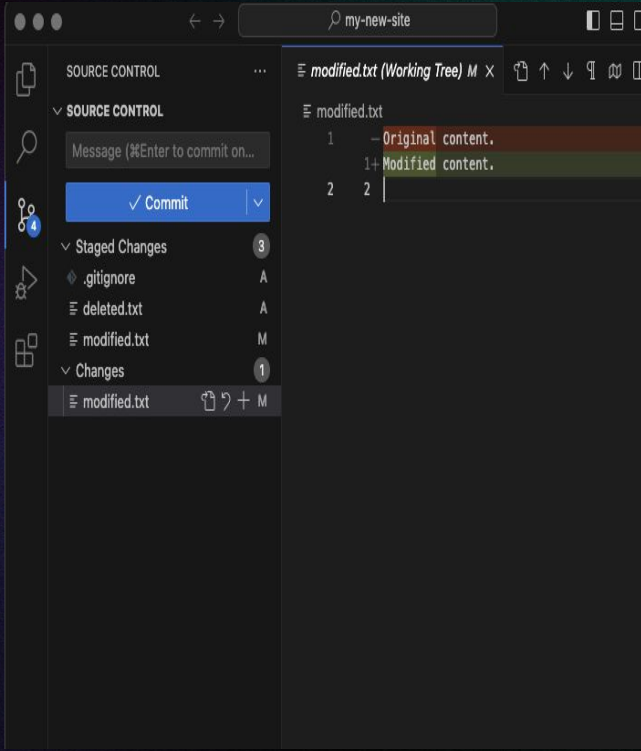


Sublime Merge



```
~ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   deleted.txt
        new file:   modified.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   modified.txt

~ 
```

Git CLI



VS Code

Note: Command Syntax

- Commands use `<>` to wrap sections of code that you should replace and `[]` to denote optional segments.
- Shorthand aliases (one `-` instead of `--`) require a space instead of an equal sign when they accept values.

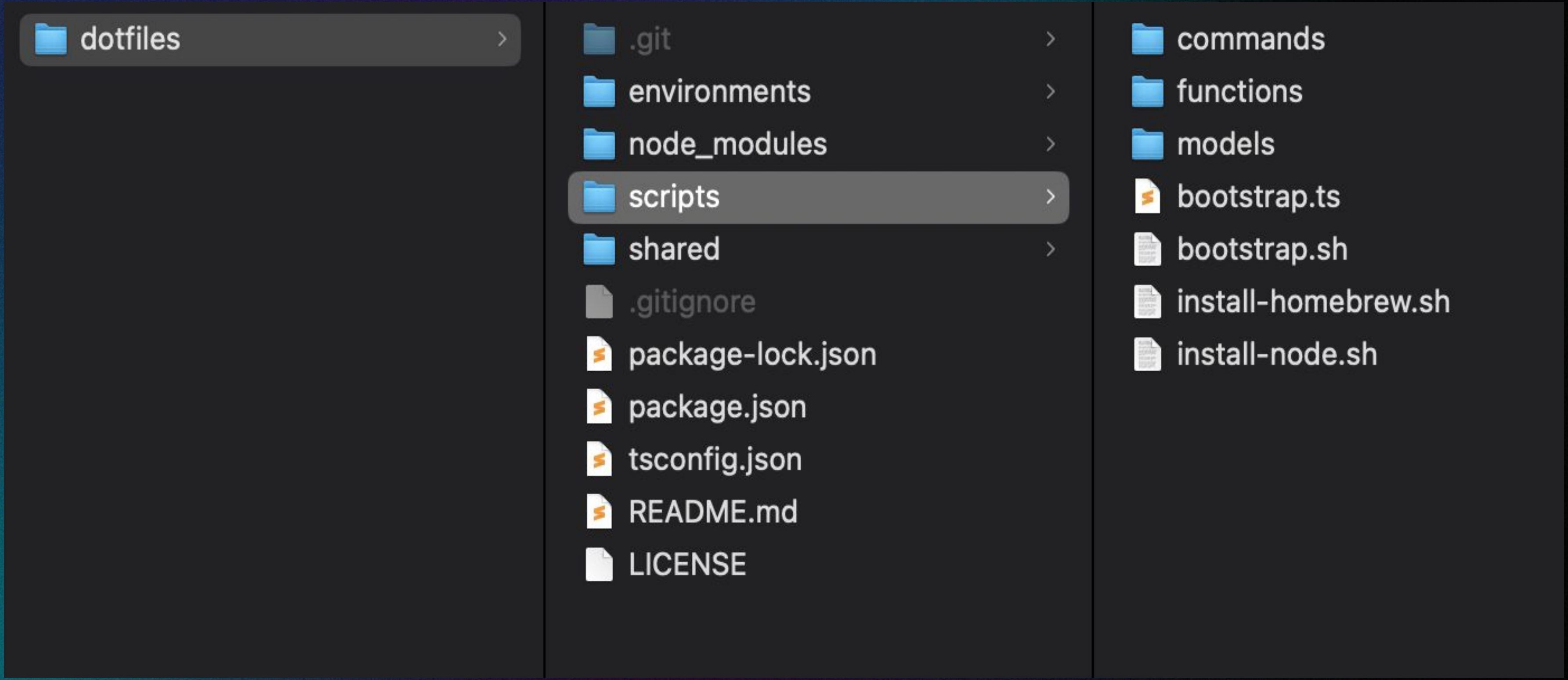
```
git log –max-count=<number>
git log -n <number>
git log -<number>
```

- Order of arguments is sometimes important. In such cases they are listed in the order they must go in.

```
git diff --staged [<path to file(s)>]
```

Term: Repository

Directory storing all Git information.



How Do I Create A New Repository?

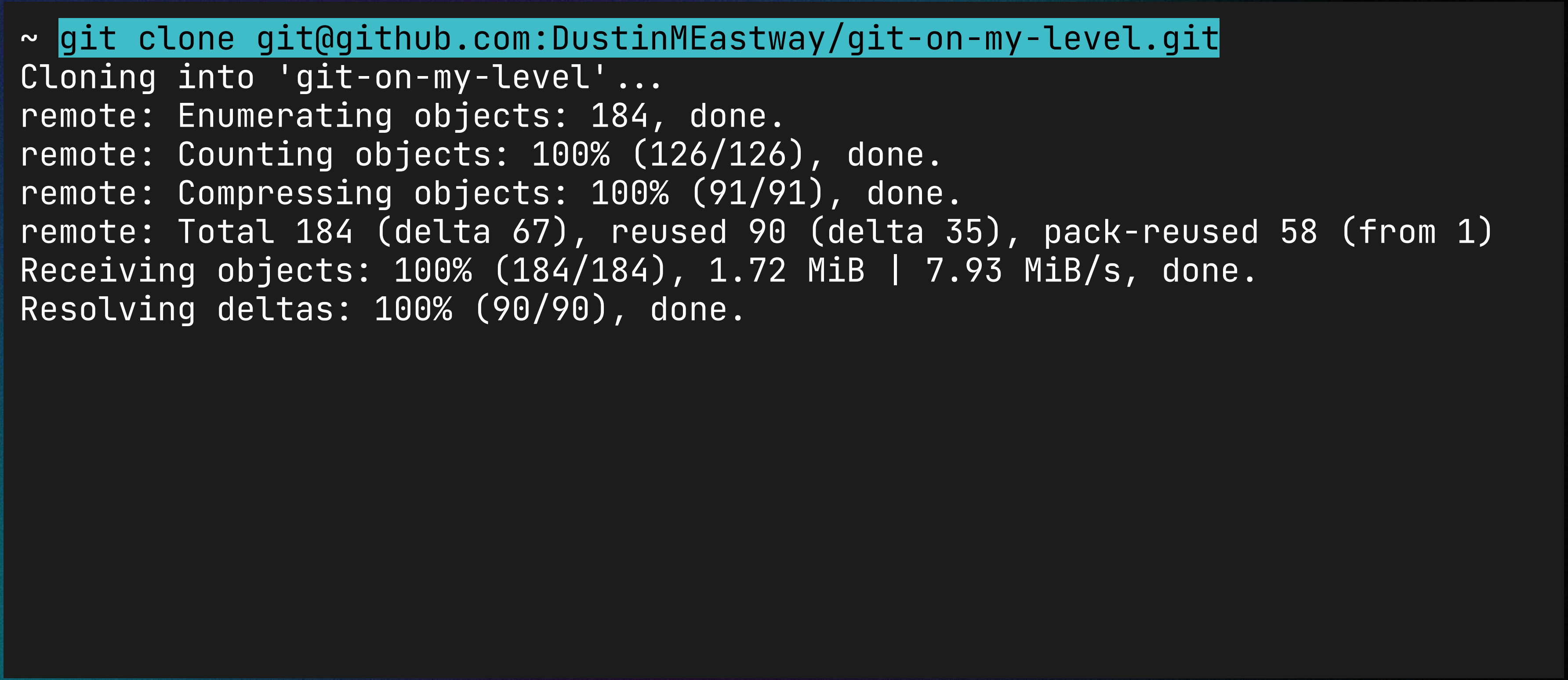
`git init`

```
~ mkdir my-new-site
my-new-site
~ cd my-new-site
~ git init
Initialized empty Git repository in /Users/deastway/Sites/my-new-site/.git/
```

How Do I Clone An Existing Repository?

Using SSH.

`git clone <ssh url>`



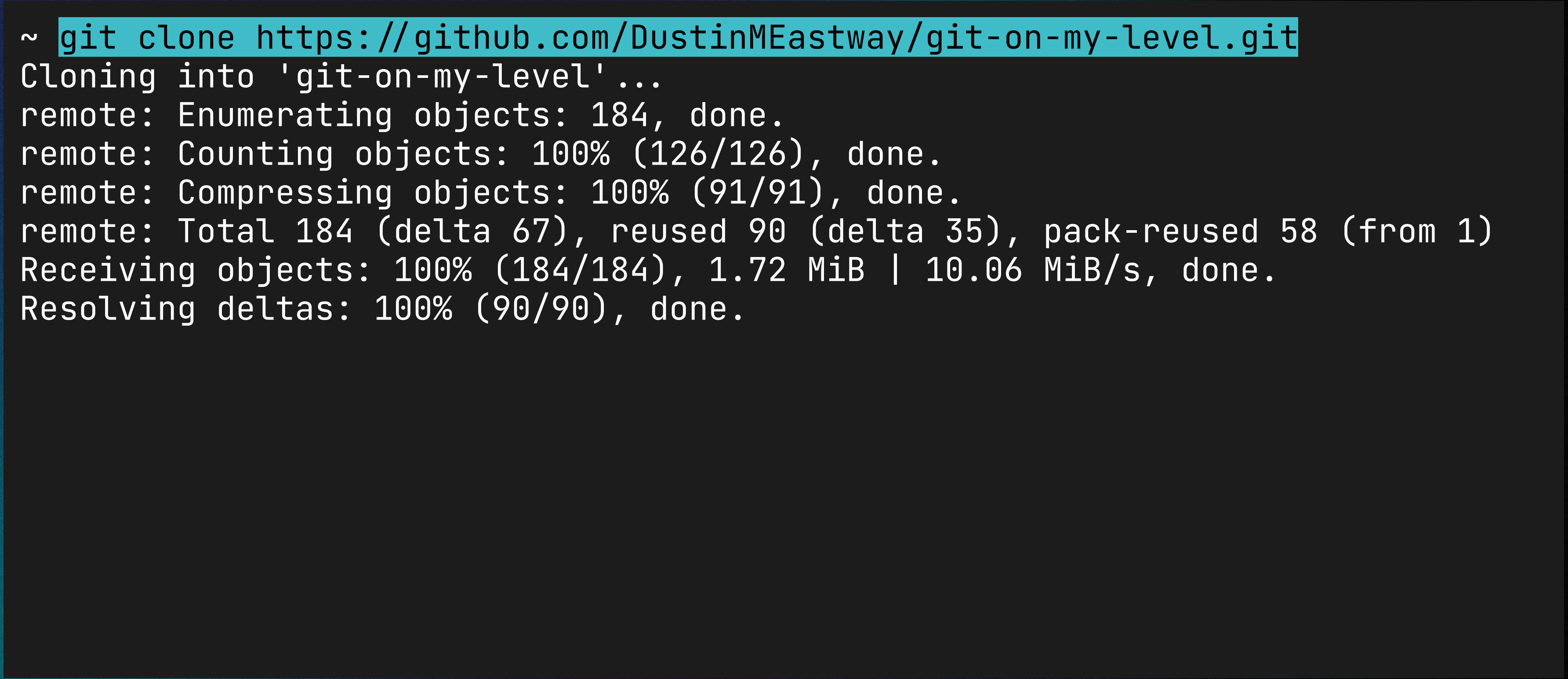
```
~ git clone git@github.com:DustinMEastway/git-on-my-level.git
Cloning into 'git-on-my-level'...
remote: Enumerating objects: 184, done.
remote: Counting objects: 100% (126/126), done.
remote: Compressing objects: 100% (91/91), done.
remote: Total 184 (delta 67), reused 90 (delta 35), pack-reused 58 (from 1)
Receiving objects: 100% (184/184), 1.72 MiB | 7.93 MiB/s, done.
Resolving deltas: 100% (90/90), done.
```

Note: Do not follow along on this slide.

How Do I Clone An Existing Repository?

Using HTTPS.

git clone <https url>



```
~ git clone https://github.com/DustinMEastway/git-on-my-level.git
Cloning into 'git-on-my-level'...
remote: Enumerating objects: 184, done.
remote: Counting objects: 100% (126/126), done.
remote: Compressing objects: 100% (91/91), done.
remote: Total 184 (delta 67), reused 90 (delta 35), pack-reused 58 (from 1)
Receiving objects: 100% (184/184), 1.72 MiB | 10.06 MiB/s, done.
Resolving deltas: 100% (90/90), done.
```

Note: Do not follow along on this slide.

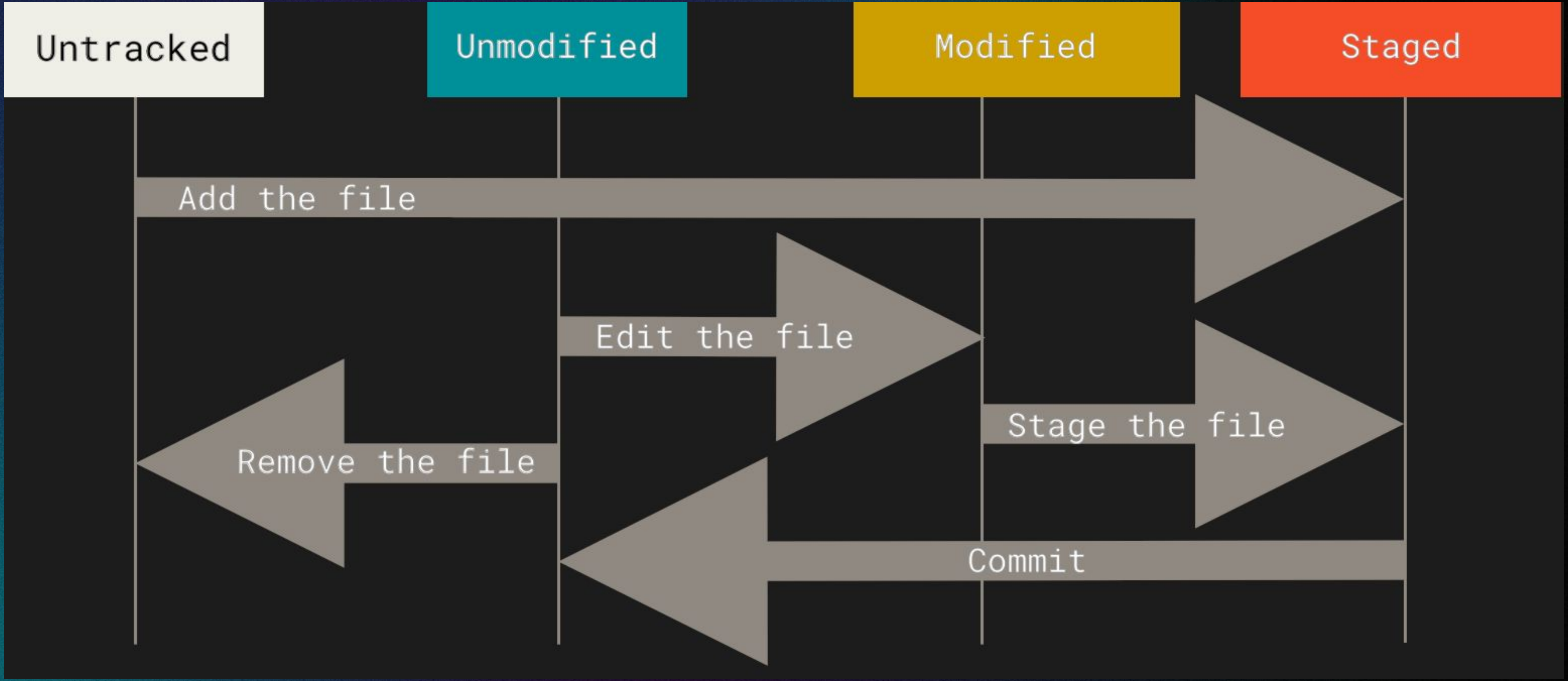
How Do I Get The Repository’s Status?

git status

```
~ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

```
~ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Note: Four States Of Files

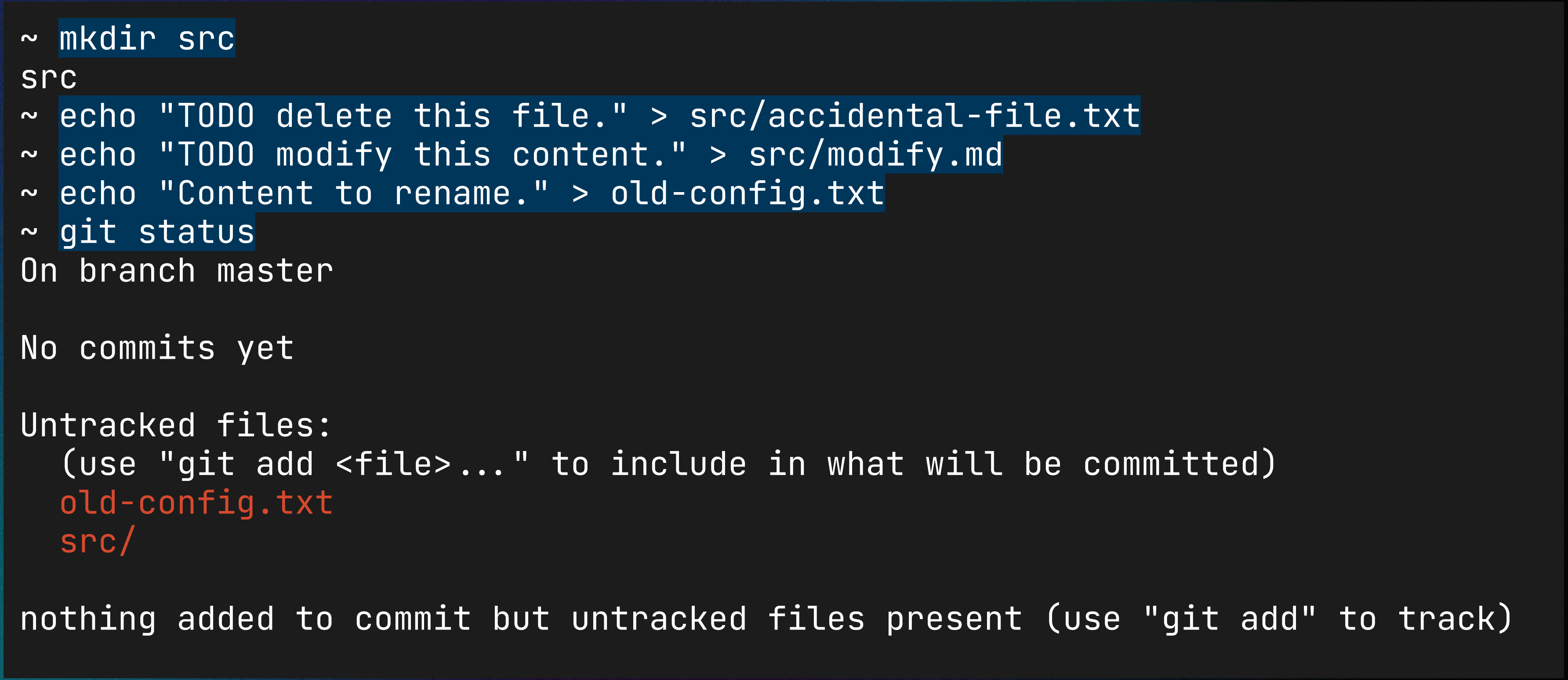


Source: Chapter 2.2 (figure 8) of the Pro Git book V2

How Do I Get The Repository’s Status?

Continued...

git status



```
~ mkdir src
src
~ echo "TODO delete this file." > src/accidental-file.txt
~ echo "TODO modify this content." > src/modify.md
~ echo "Content to rename." > old-config.txt
~ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  old-config.txt
  src/

nothing added to commit but untracked files present (use "git add" to track)
```

How Do I Add Files To A Repository?

File path(s).

`git add <path to file(s)>`

```
~ git add old-config.txt src/accidental-file.txt
~ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   old-config.txt
  new file:   src/accidental-file.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  src/modify.md
```

Note: File(s) Command Arguments

Pathspec - directory/file path, pattern, etc.

Accept one or more of the following separated by spaces.

- Directory path.
  - Example: `src/` matches all files in the "src" subdirectory.
- File path.
  - Example: `src/accidental-file.txt` matches "accidental-file.txt" in "src".
- Pattern.
  - Example: `*.txt` matches all ".txt" files.
  - Example: `src/*.txt` matches all ".txt" files in the "src" directory.

How Do I Add Files To A Repository?

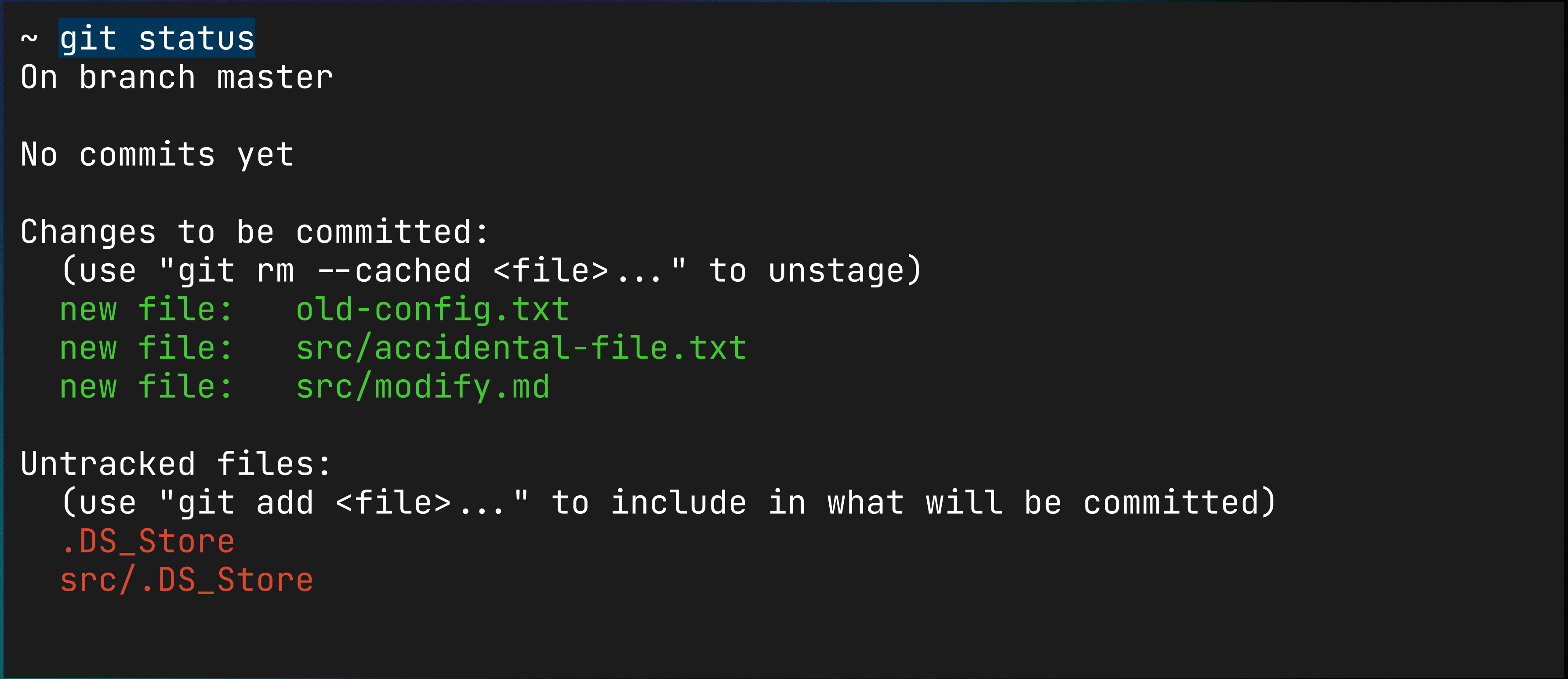
Directory path(s).

`git add <path to directory(s)>`

```
~ git add src
~ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   old-config.txt
  new file:   src/accidental-file.txt
  new file:   src/modify.md
```

How Do I Ignore Files?

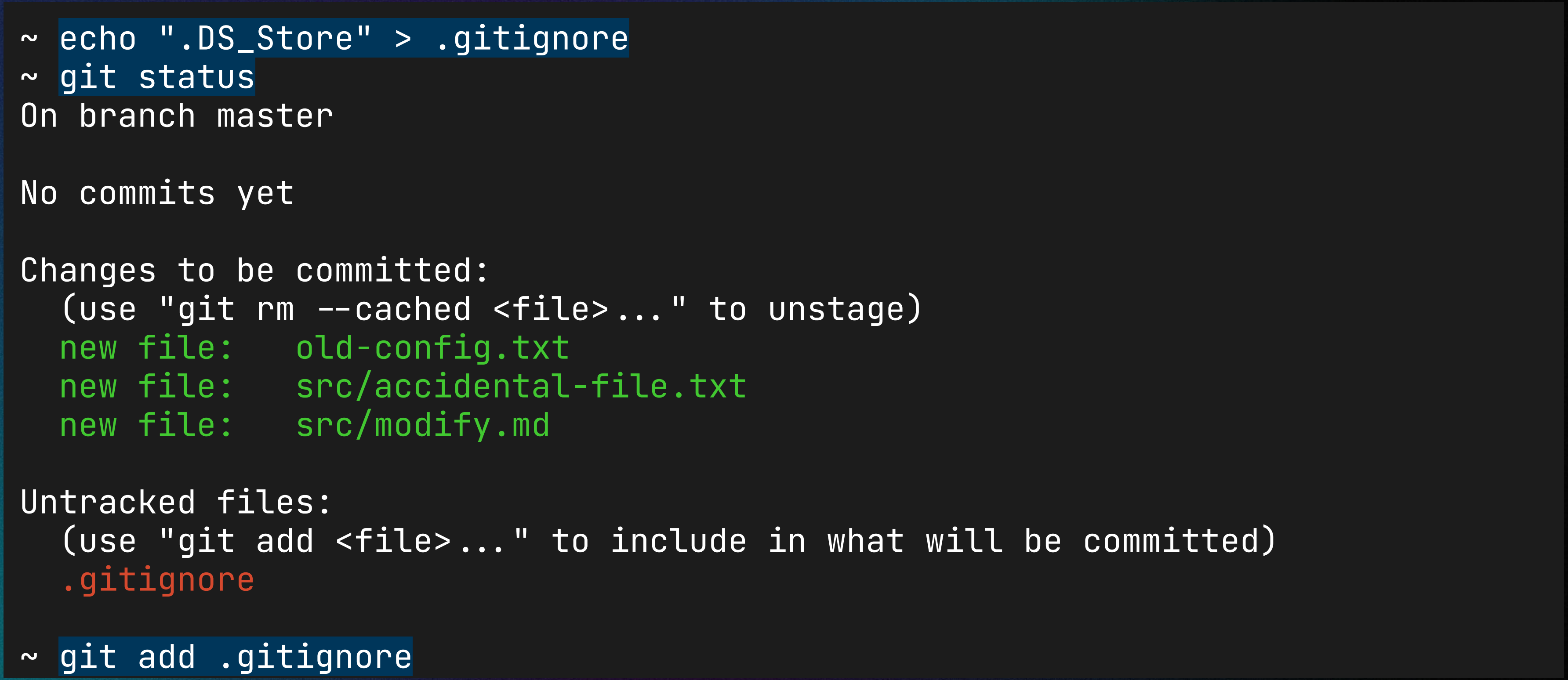
Untracked files.
Add .gitignore patterns.



```
~ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   old-config.txt
  new file:   src/accidental-file.txt
  new file:   src/modify.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  .DS_Store
  src/.DS_Store
```

How Do I Ignore Files?

Untracked files. Continued...
Add .gitignore patterns.



```
~ echo ".DS_Store" > .gitignore
~ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   old-config.txt
  new file:   src/accidental-file.txt
  new file:   src/modify.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  .gitignore

~ git add .gitignore
```

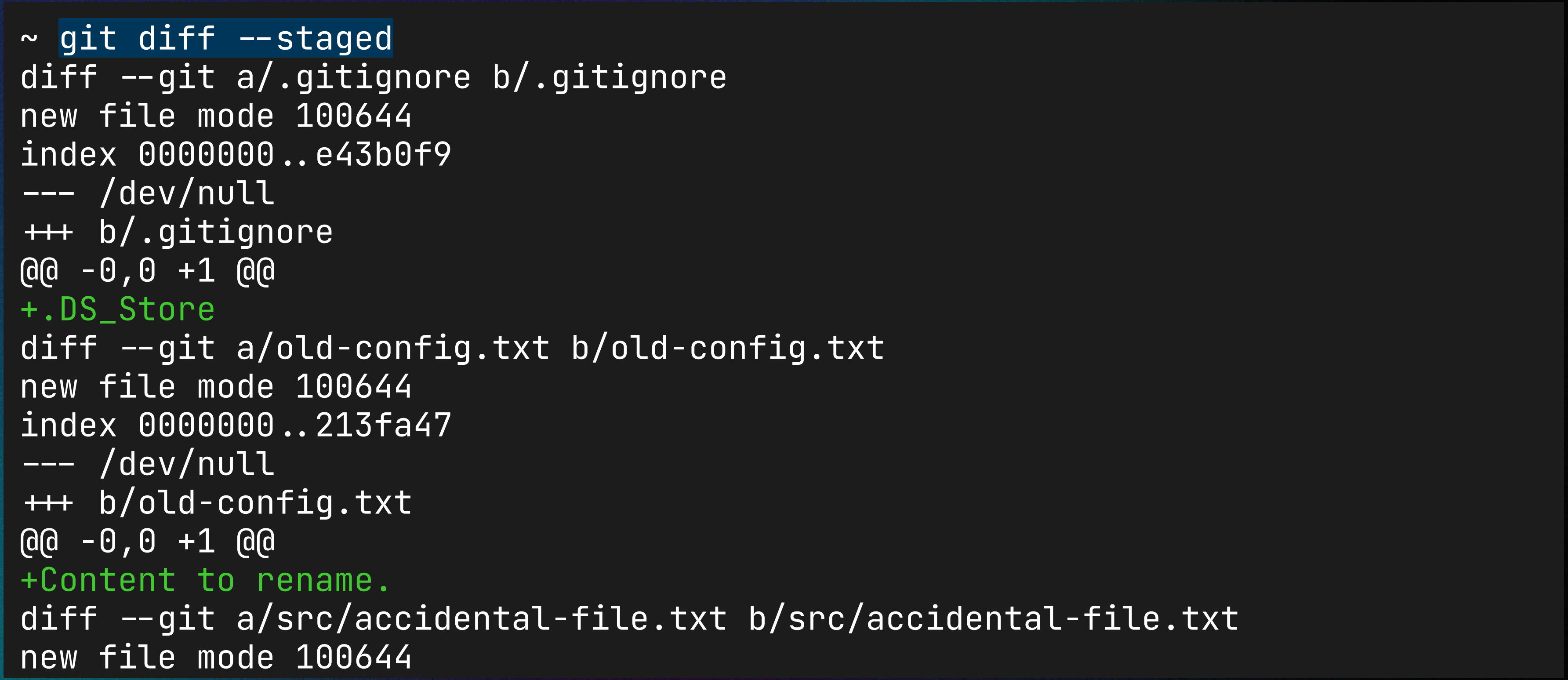
Note: Commonly Ignored Files

- Dependencies (e.g. “node_modules”).
- Compiled output (e.g. “dist”, “build”, etc.).
- Files with secrets (e.g. “.env”).
- Personal IDE configurations (e.g. “.vscode”).

How Do I Display Changes?

Staged changes.

`git diff --staged`

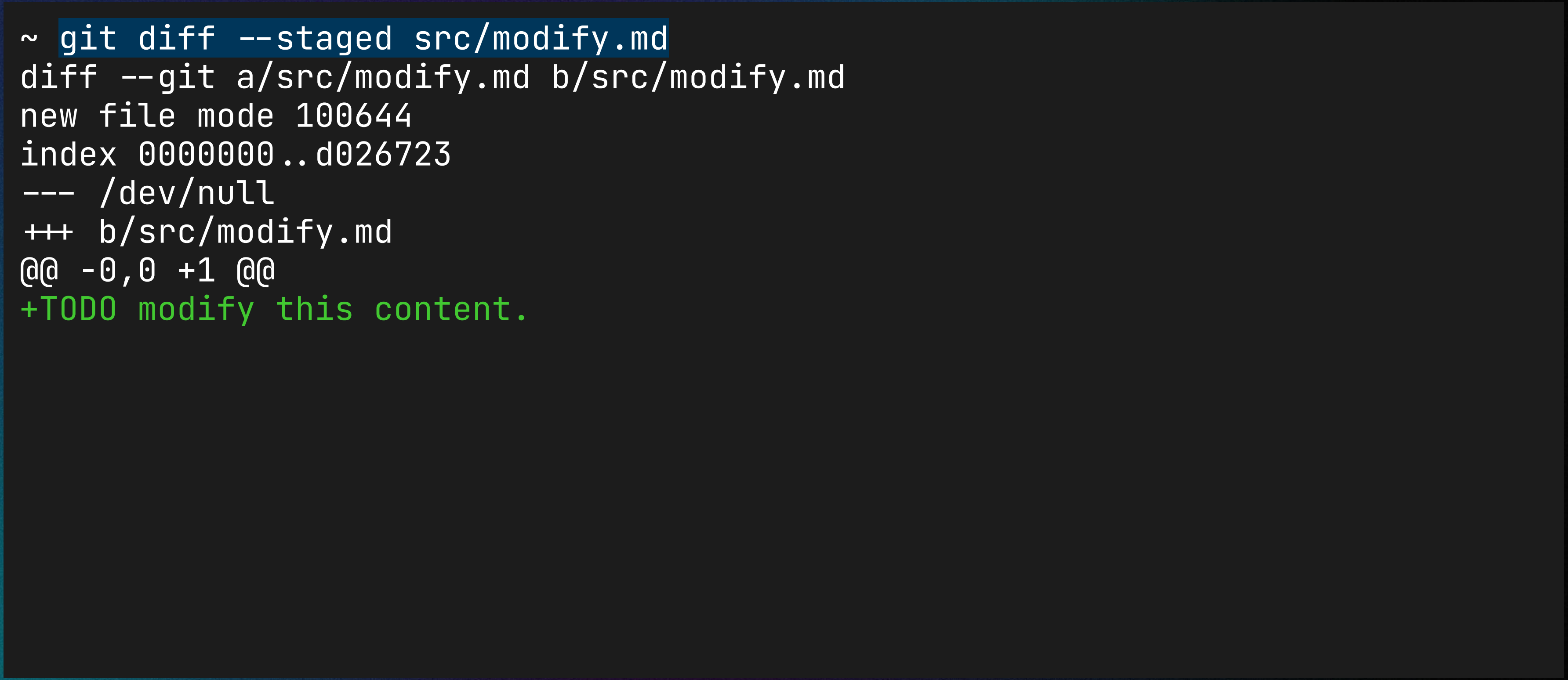


```
~ git diff --staged
diff --git a/.gitignore b/.gitignore
new file mode 100644
index 0000000..e43b0f9
--- /dev/null
+++ b/.gitignore
@@ -0,0 +1 @@
+.DS_Store
diff --git a/old-config.txt b/old-config.txt
new file mode 100644
index 0000000..213fa47
--- /dev/null
+++ b/old-config.txt
@@ -0,0 +1 @@
+Content to rename.
diff --git a/src/accidental-file.txt b/src/accidental-file.txt
new file mode 100644
```

How Do I Display Changes?

Staged changes in specific file(s).

`git diff --staged <path to file(s)>`



```
~ git diff --staged src/modify.md
diff --git a/src/modify.md b/src/modify.md
new file mode 100644
index 0000000..d026723
--- /dev/null
+++ b/src/modify.md
@@ -0,0 +1 @@
+TODO modify this content.
```

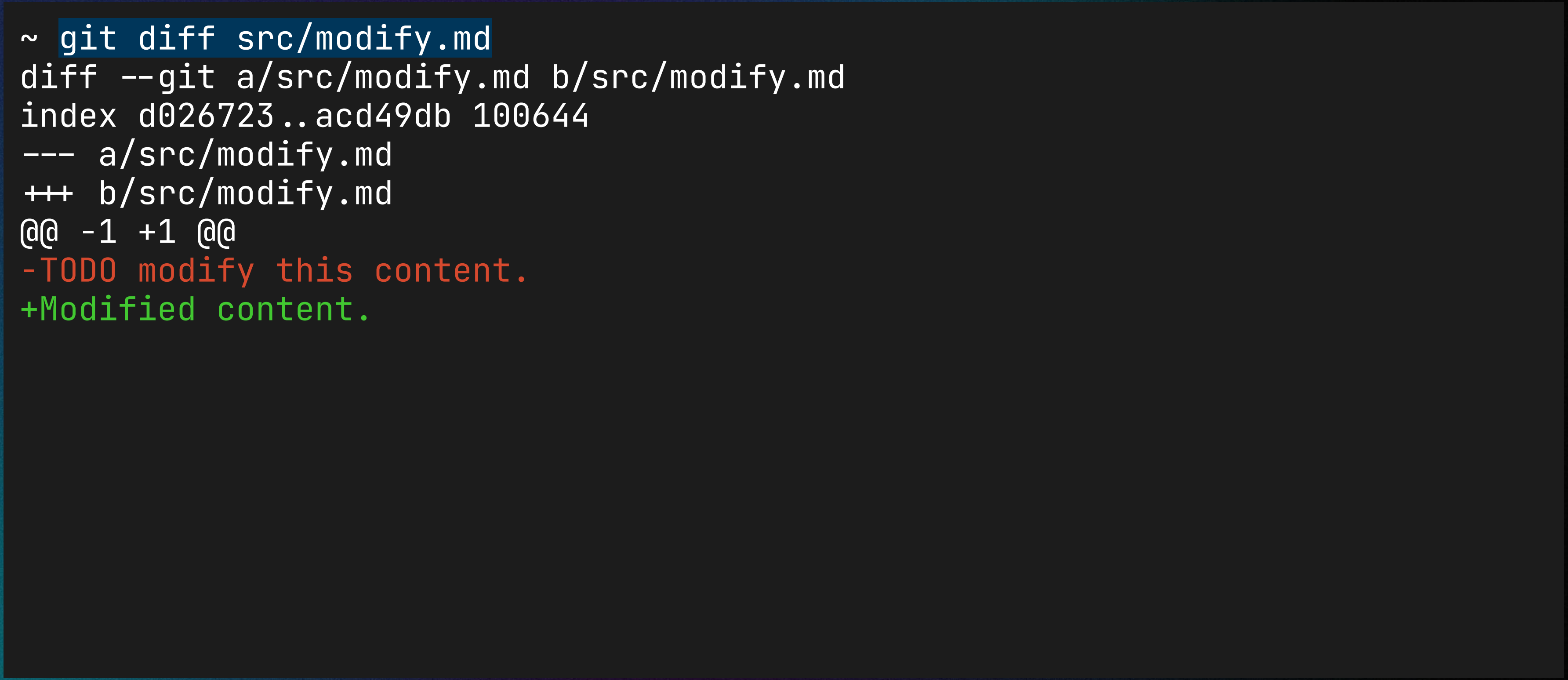
Note: Modifying A Staged File

```
~ echo "Modified content." > src/modify.md
~ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   .gitignore
  new file:   old-config.txt
  new file:   src/accidental-file.txt
  new file:   src/modify.md

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
  modified:   src/modify.md
```

How Do I Display Changes?

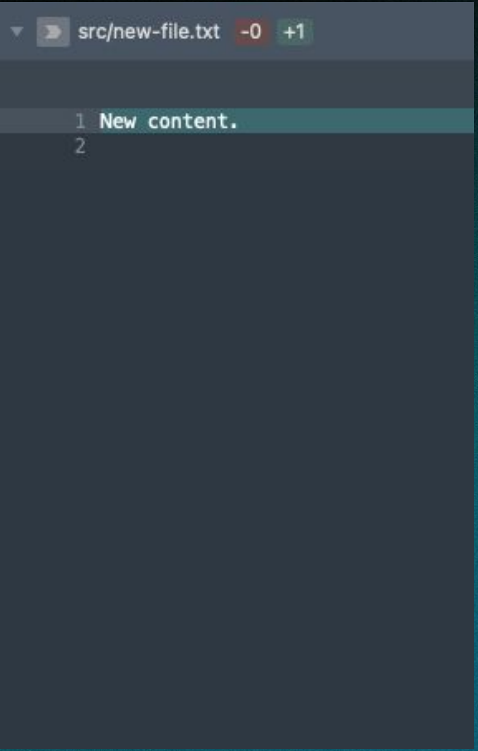
Unstaged changes.

git diff [<path to file(s)>]



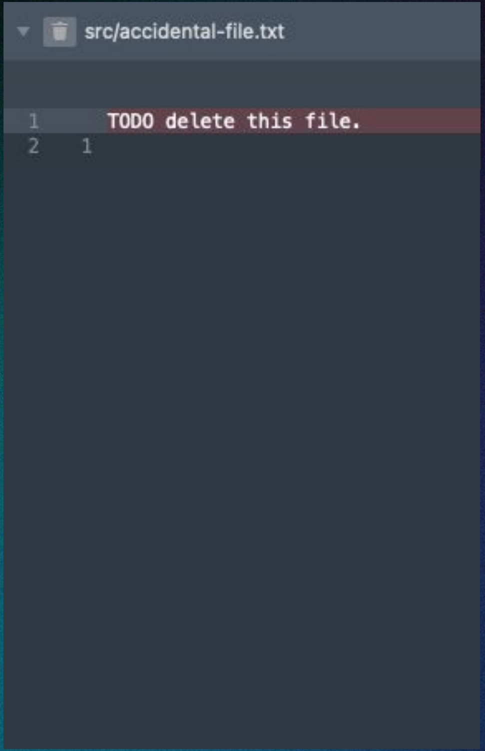
```
~ git diff src/modify.md
diff --git a/src/modify.md b/src/modify.md
index d026723..acd49db 100644
--- a/src/modify.md
+++ b/src/modify.md
@@ -1 +1 @@
-TODO modify this content.
+Modified content.
```

Term: Change



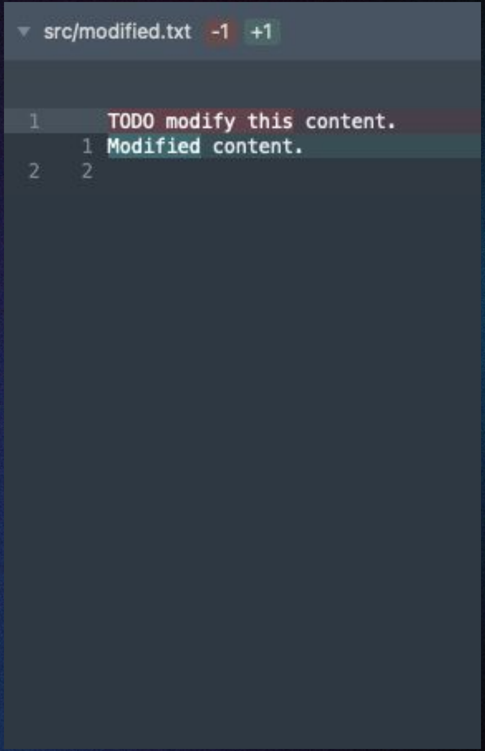
```
src/new-file.txt -0 +1

New content.
```

Add a file.



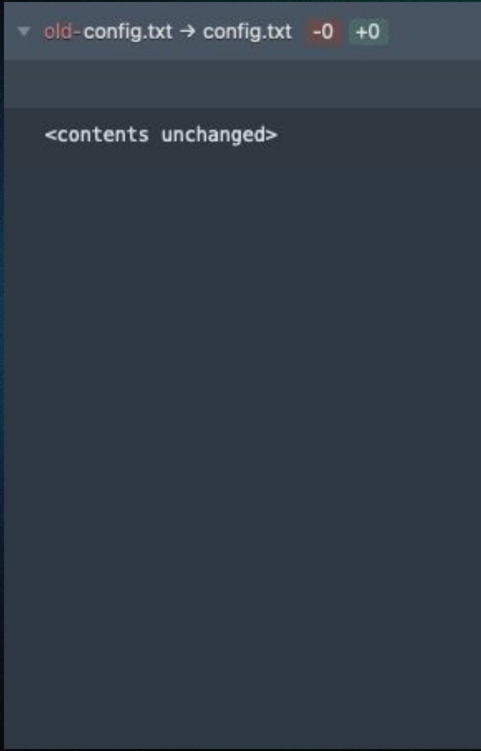
```
src/accidental-file.txt

TODO delete this file.
```

Delete a file.



```
src/modified.txt -1 +1

TODO modify this content.
Modified content.
```

Edit lines.



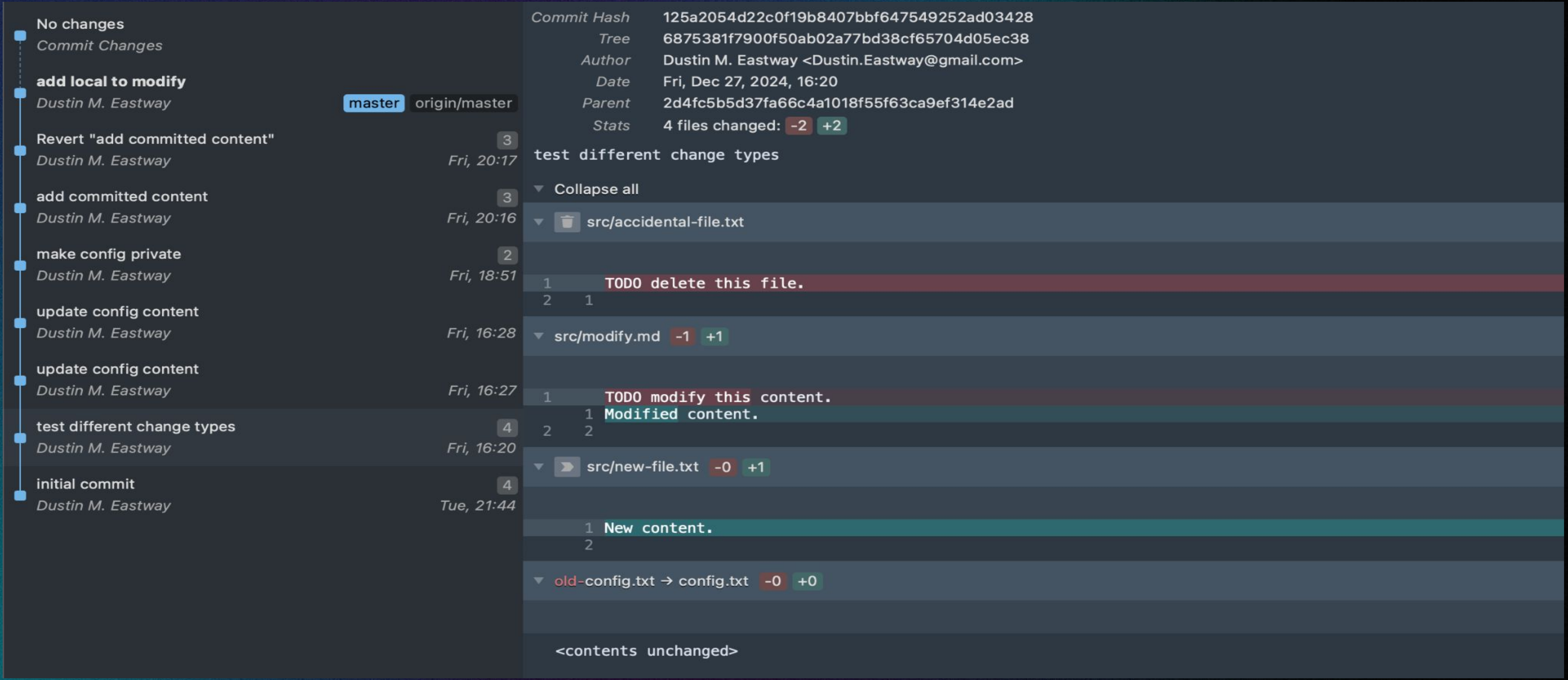
```
old-config.txt → config.txt -0 +0

<contents unchanged>
```

Move/rename a file.

Note: Four States Of Changes

- **Untracked** - Newly created untracked file.
- **Unstaged** - Change made to an unmodified or staged file that has not been staged yet.
- **Staged** - Change will go into next commit.
- **Committed** - Change already added to a commit to “save” it.

Term: Commit



No changes
Commit Changes

add local to modify
Dustin M. Eastway master origin/master

Revert "add committed content" 3
Dustin M. Eastway *Fri, 20:17*

add committed content 3
Dustin M. Eastway *Fri, 20:16*

make config private 2
Dustin M. Eastway *Fri, 18:51*

update config content
Dustin M. Eastway *Fri, 16:28*

update config content
Dustin M. Eastway *Fri, 16:27*

test different change types 4
Dustin M. Eastway *Fri, 16:20*

initial commit 4
Dustin M. Eastway *Tue, 21:44*

```
Commit Hash  125a2054d22c0f19b8407bbf647549252ad03428
       Tree  6875381f7900f50ab02a77bd38cf65704d05ec38
     Author  Dustin M. Eastway <Dustin.Eastway@gmail.com>
       Date  Fri, Dec 27, 2024, 16:20
     Parent  2d4fc5b5d37fa66c4a1018f55f63ca9ef314e2ad
      Stats  4 files changed: -2 +2
test different change types
```

Collapse all

src/accidental-file.txt

```
1       TODO delete this file.
2    1
```

src/modify.md -1 +1

```
1       TODO modify this content.
     1  Modified content.
2    2
```

src/new-file.txt -0 +1

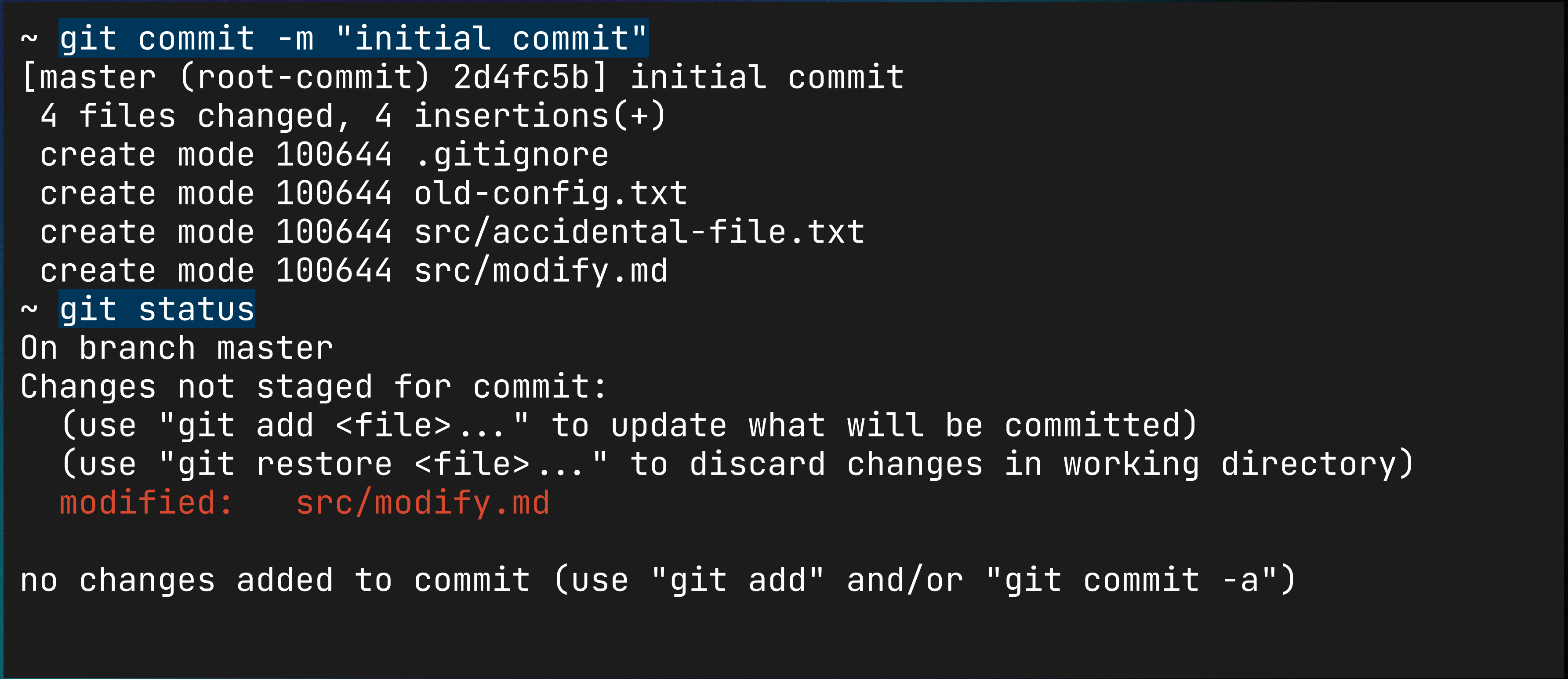
```
     1  New content.
     2
```

old-config.txt → config.txt -0 +0

```
<contents unchanged>
```

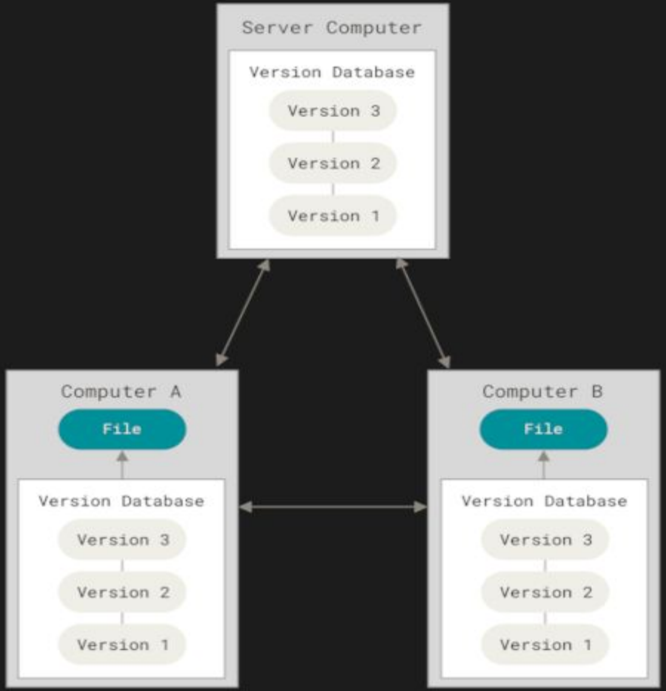
How Do I Save Changes?

`git commit [-m "<commit message>"]`



```
~ git commit -m "initial commit"
[master (root-commit) 2d4fc5b] initial commit
 4 files changed, 4 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 old-config.txt
 create mode 100644 src/accidental-file.txt
 create mode 100644 src/modify.md
~ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
	modified:   src/modify.md

no changes added to commit (use "git add" and/or "git commit -a")
```

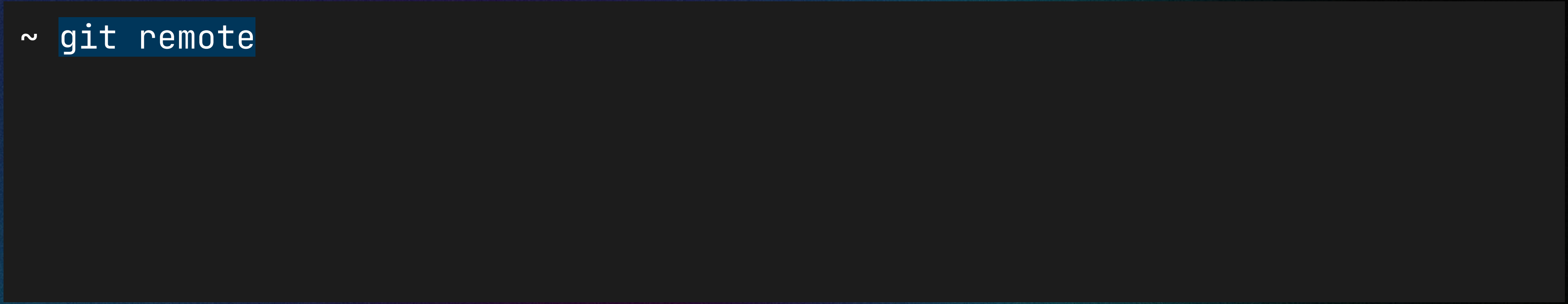
Term: Local & Remote



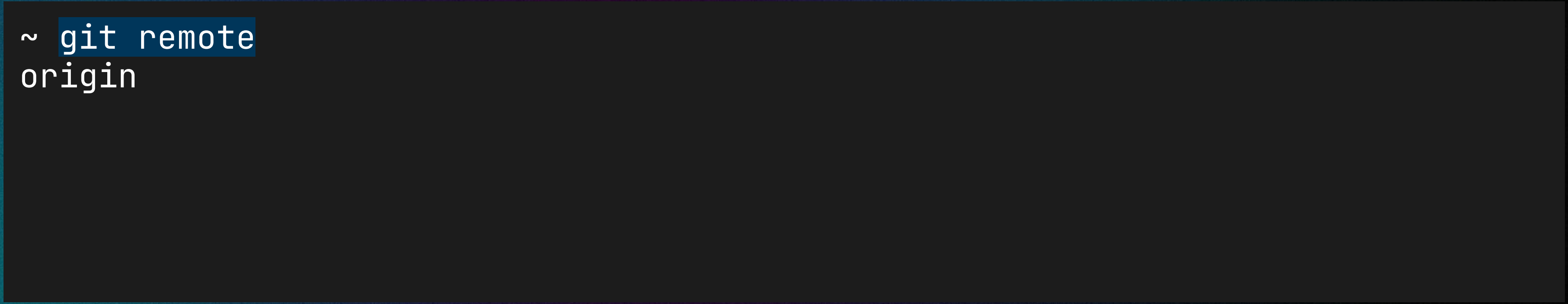
Source: Chapter 1.1 (figure 3) of the Pro Git book V2

How Do I Display Remotes?

`git remote`



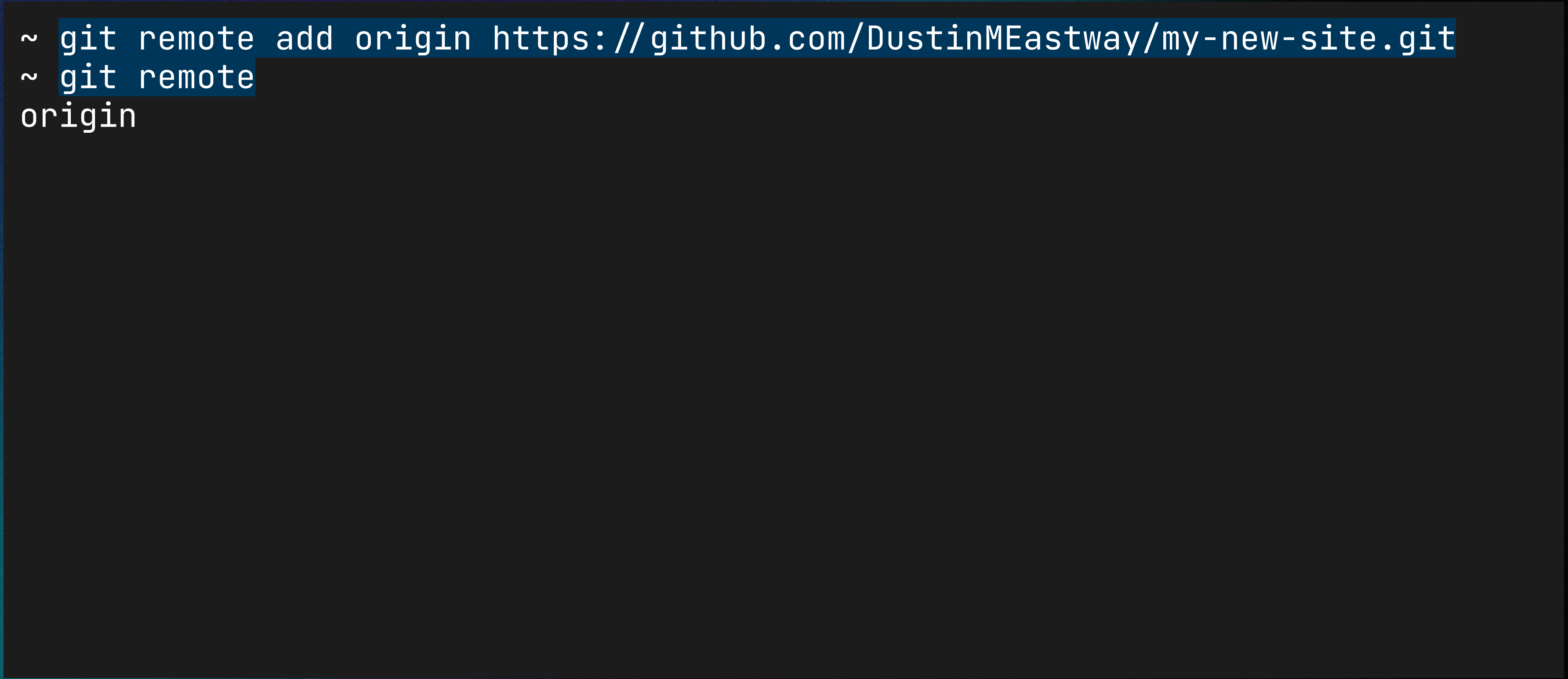
```
~ git remote
```



```
~ git remote
origin
```

How Do I Add A Remote?

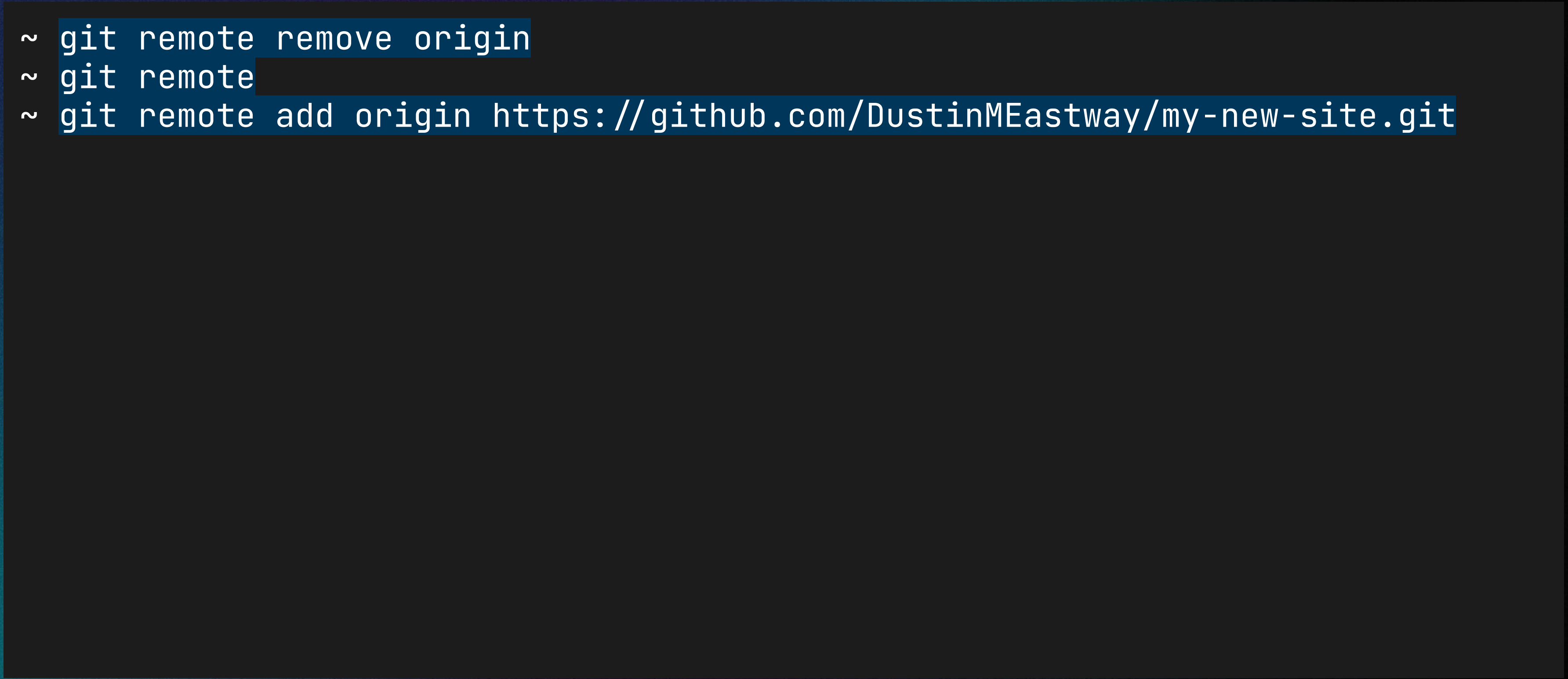
git remote add <remote name> <remote url>



```
~ git remote add origin https://github.com/DustinMEastway/my-new-site.git
~ git remote
origin
```

How Do I Remove A Remote?

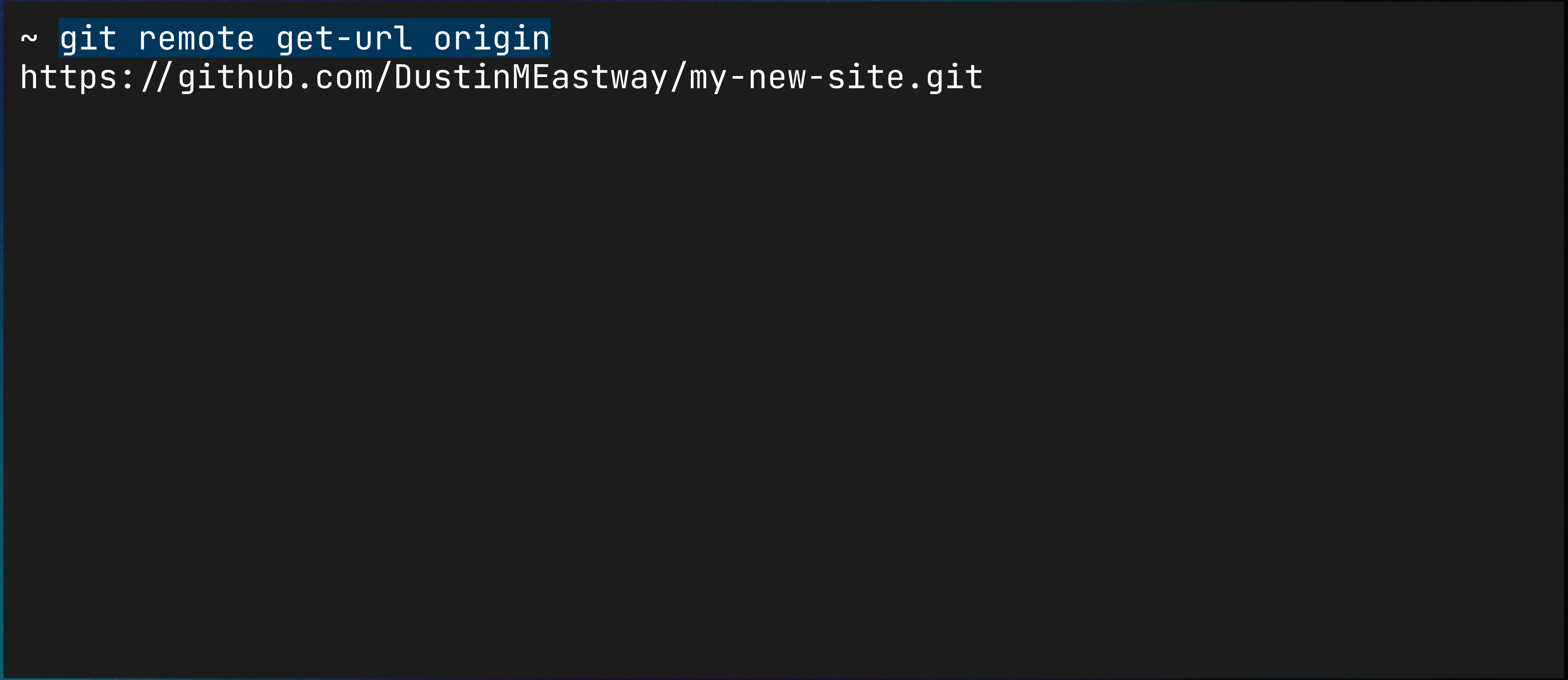
git remote remove <remote name>



```
~ git remote remove origin
~ git remote
~ git remote add origin https://github.com/DustinMEastway/my-new-site.git
```

How Do I Display The URL For A Remote?

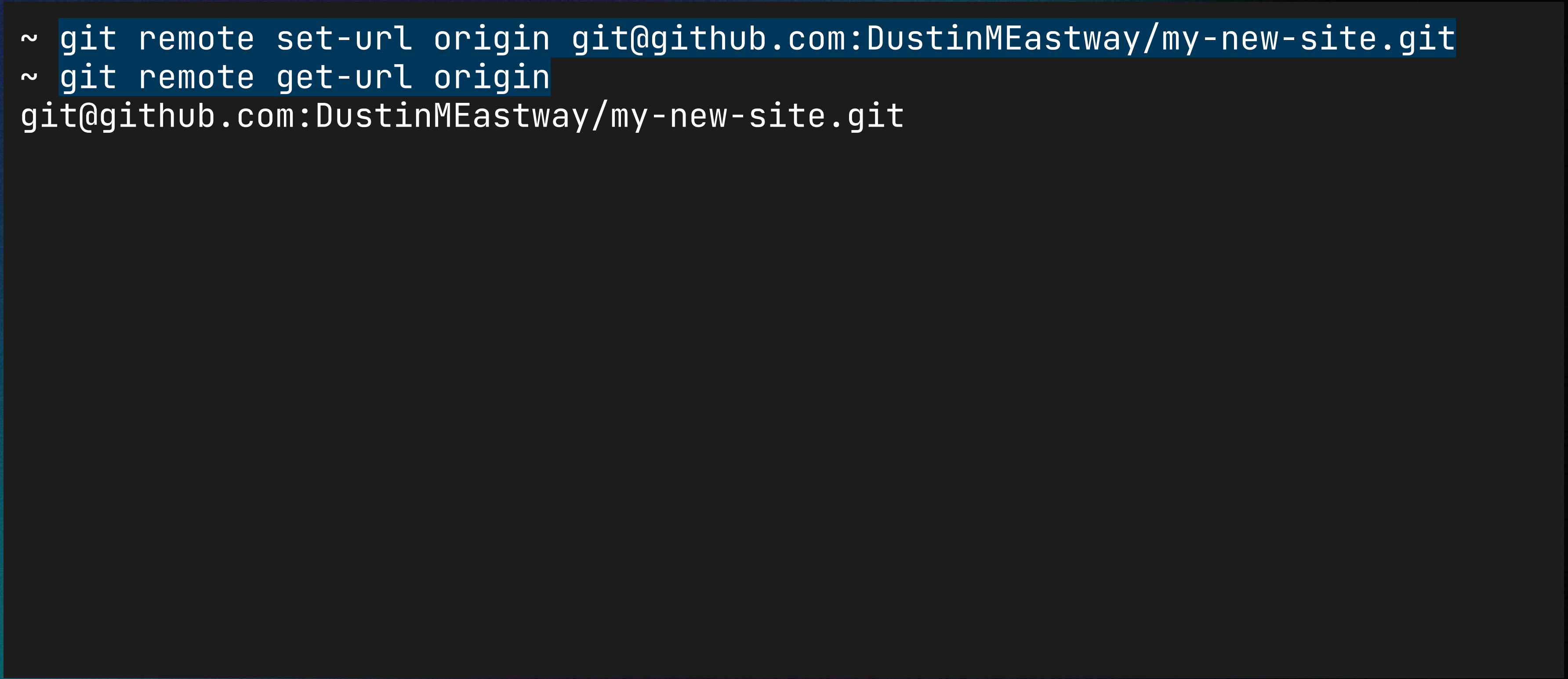
git remote get-url <remote name>



```
~ git remote get-url origin
https://github.com/DustinMEastway/my-new-site.git
```

How Do I Set The URL For A Remote?

`git remote set-url <remote name> <remote url>`

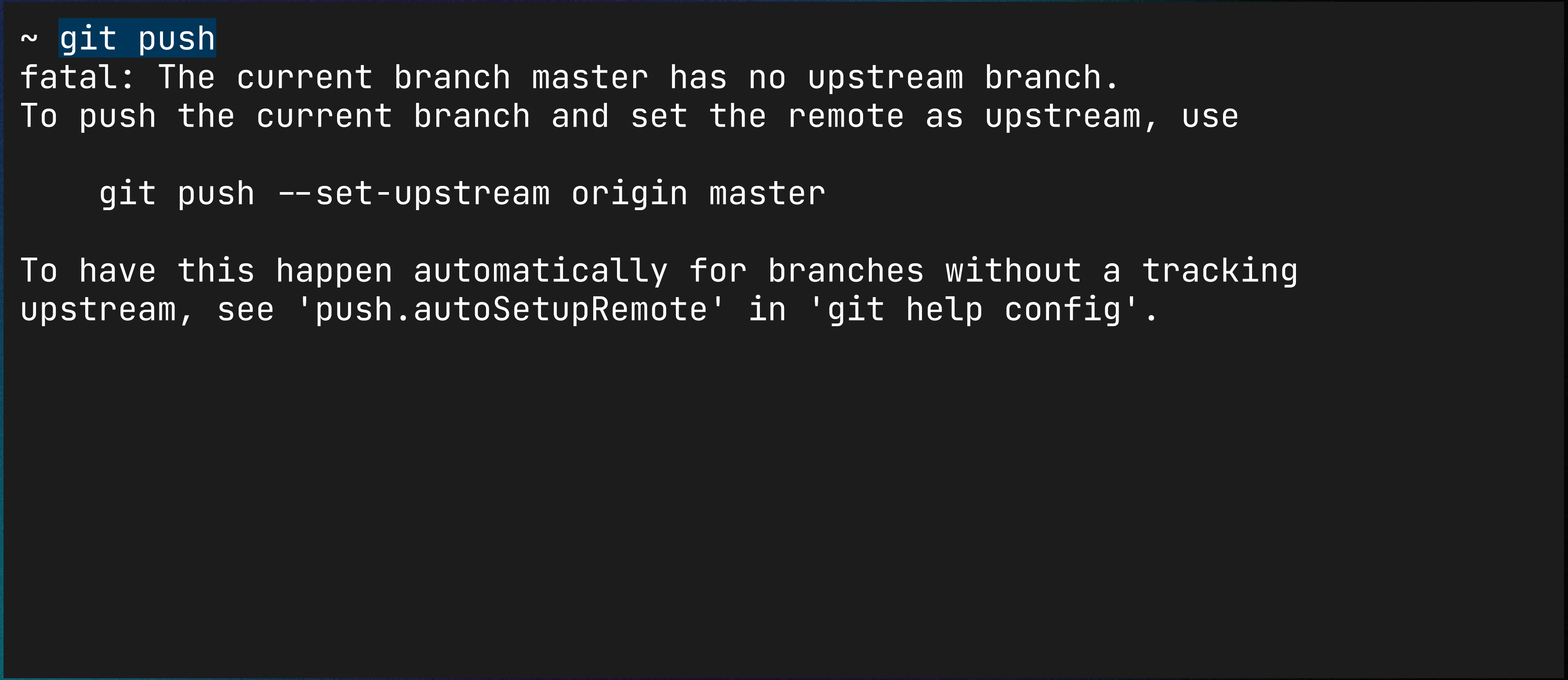


```
~ git remote set-url origin git@github.com:DustinMEastway/my-new-site.git
~ git remote get-url origin
git@github.com:DustinMEastway/my-new-site.git
```

How Do I Upload Changes To Remote?

With an unset upstream branch.

`git push --set-upstream <remote name> <branch name>`

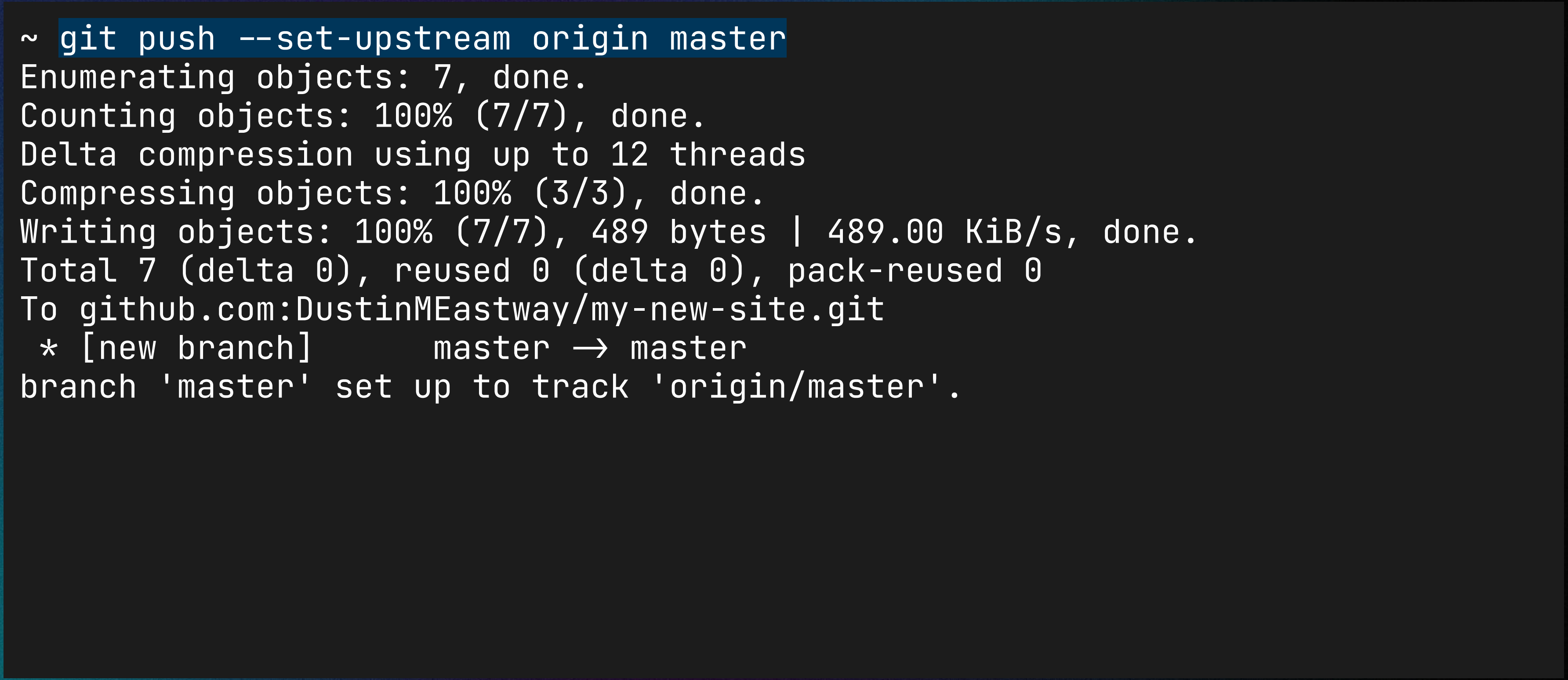


```
~ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
```

How Do I Upload Changes To Remote?

With an unset upstream branch. Continued...

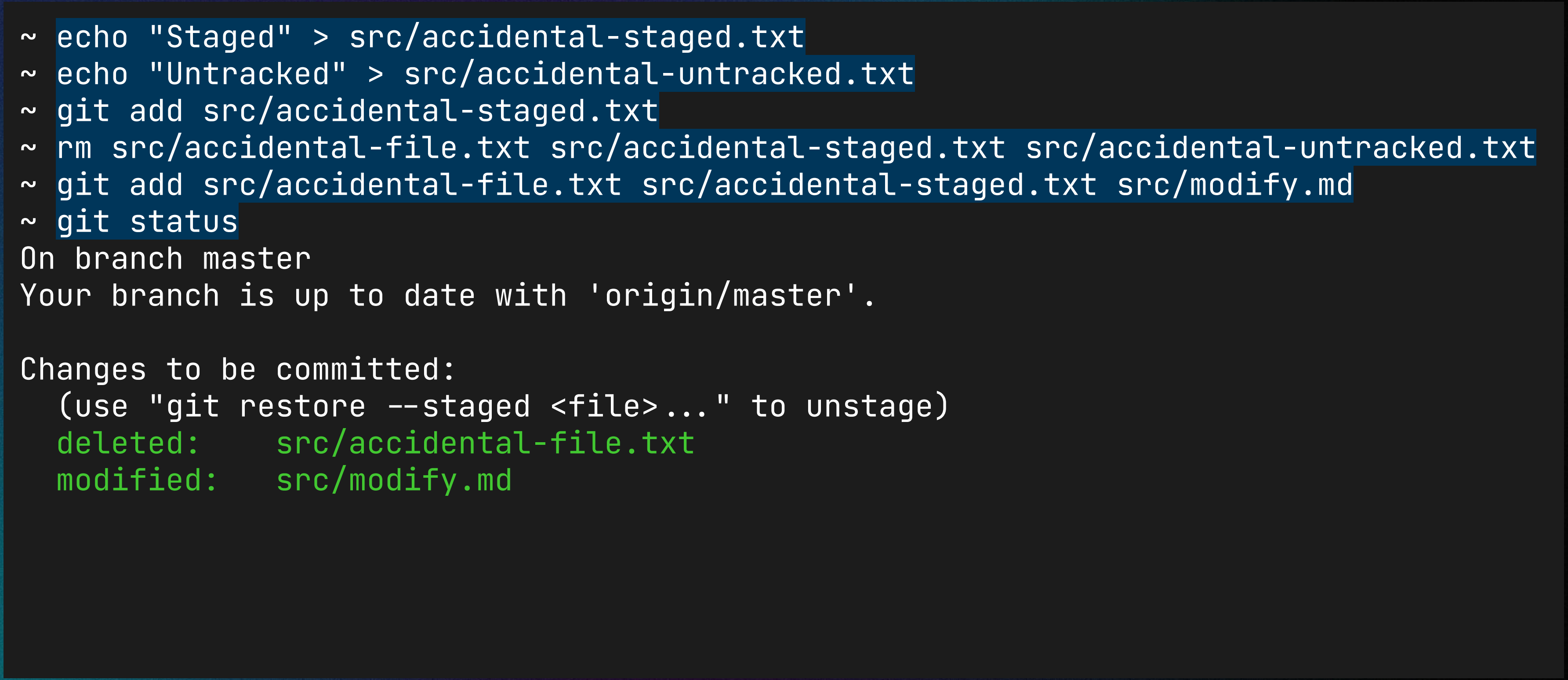
git push --set-upstream <remote name> <branch name>



```
~ git push --set-upstream origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (7/7), 489 bytes | 489.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:DustinMEastway/my-new-site.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

How Do I Delete Files?

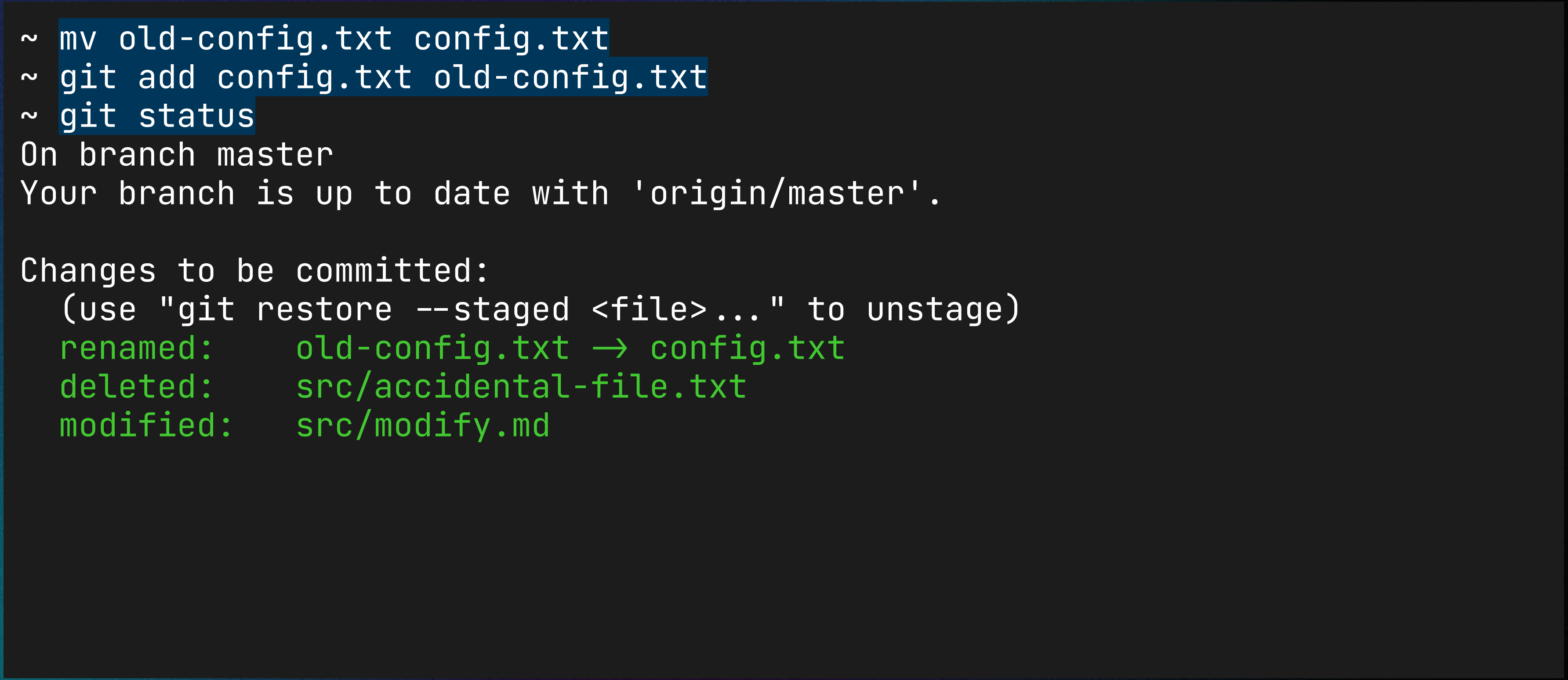
git add <path to file(s)>



```
~ echo "Staged" > src/accidental-staged.txt
~ echo "Untracked" > src/accidental-untracked.txt
~ git add src/accidental-staged.txt
~ rm src/accidental-file.txt src/accidental-staged.txt src/accidental-untracked.txt
~ git add src/accidental-file.txt src/accidental-staged.txt src/modify.md
~ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  deleted:    src/accidental-file.txt
  modified:   src/modify.md
```

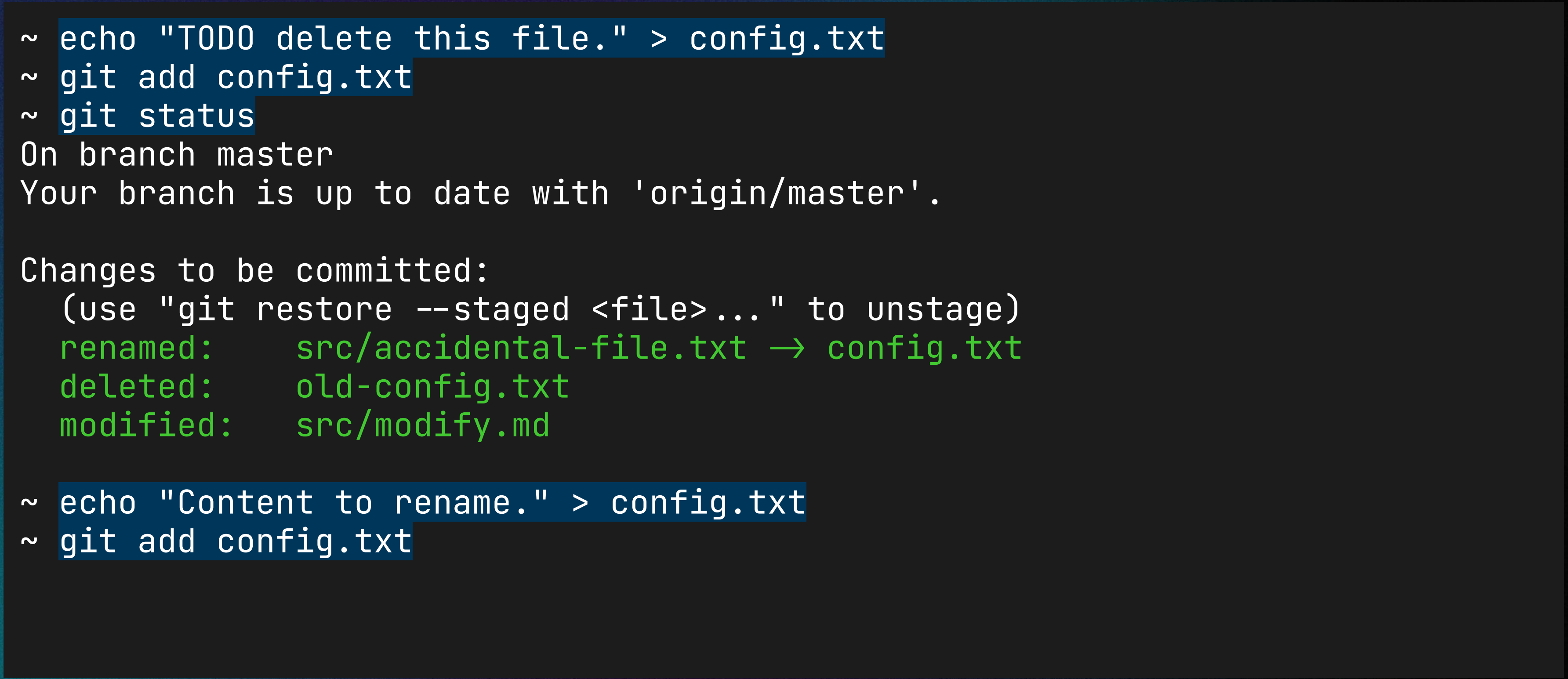
How Do I Move / Rename Files?

git add <path to file(s)>



```
~ mv old-config.txt config.txt
~ git add config.txt old-config.txt
~ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  renamed:    old-config.txt -> config.txt
  deleted:    src/accidental-file.txt
  modified:   src/modify.md
```

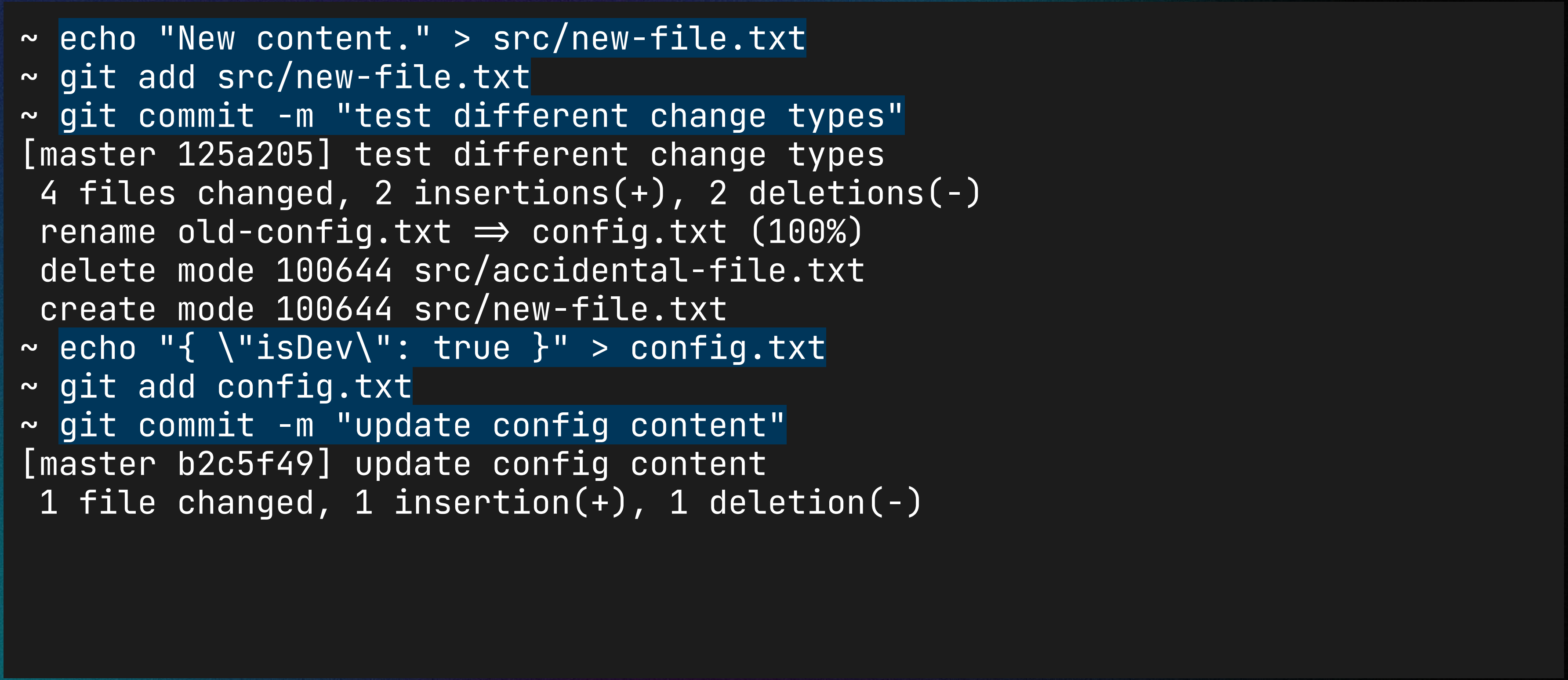
Warning: Dangers Of Moving Files



```
~ echo "TODO delete this file." > config.txt
~ git add config.txt
~ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  renamed:    src/accidental-file.txt -> config.txt
  deleted:    old-config.txt
  modified:   src/modify.md

~ echo "Content to rename." > config.txt
~ git add config.txt
```

Prep Content

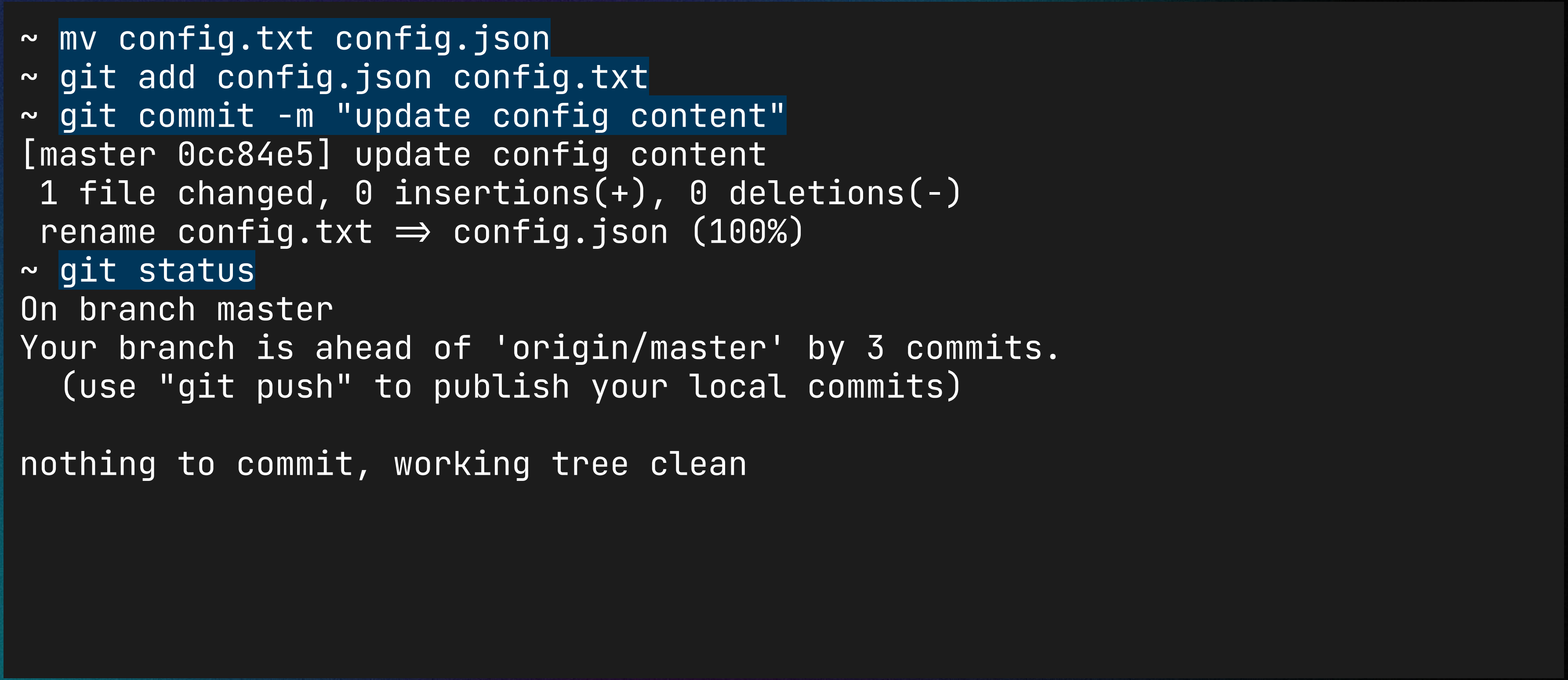
Add commits to display.



```
~ echo "New content." > src/new-file.txt
~ git add src/new-file.txt
~ git commit -m "test different change types"
[master 125a205] test different change types
 4 files changed, 2 insertions(+), 2 deletions(-)
 rename old-config.txt => config.txt (100%)
 delete mode 100644 src/accidental-file.txt
 create mode 100644 src/new-file.txt
~ echo "{ \"isDev\": true }" > config.txt
~ git add config.txt
~ git commit -m "update config content"
[master b2c5f49] update config content
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Prep Content

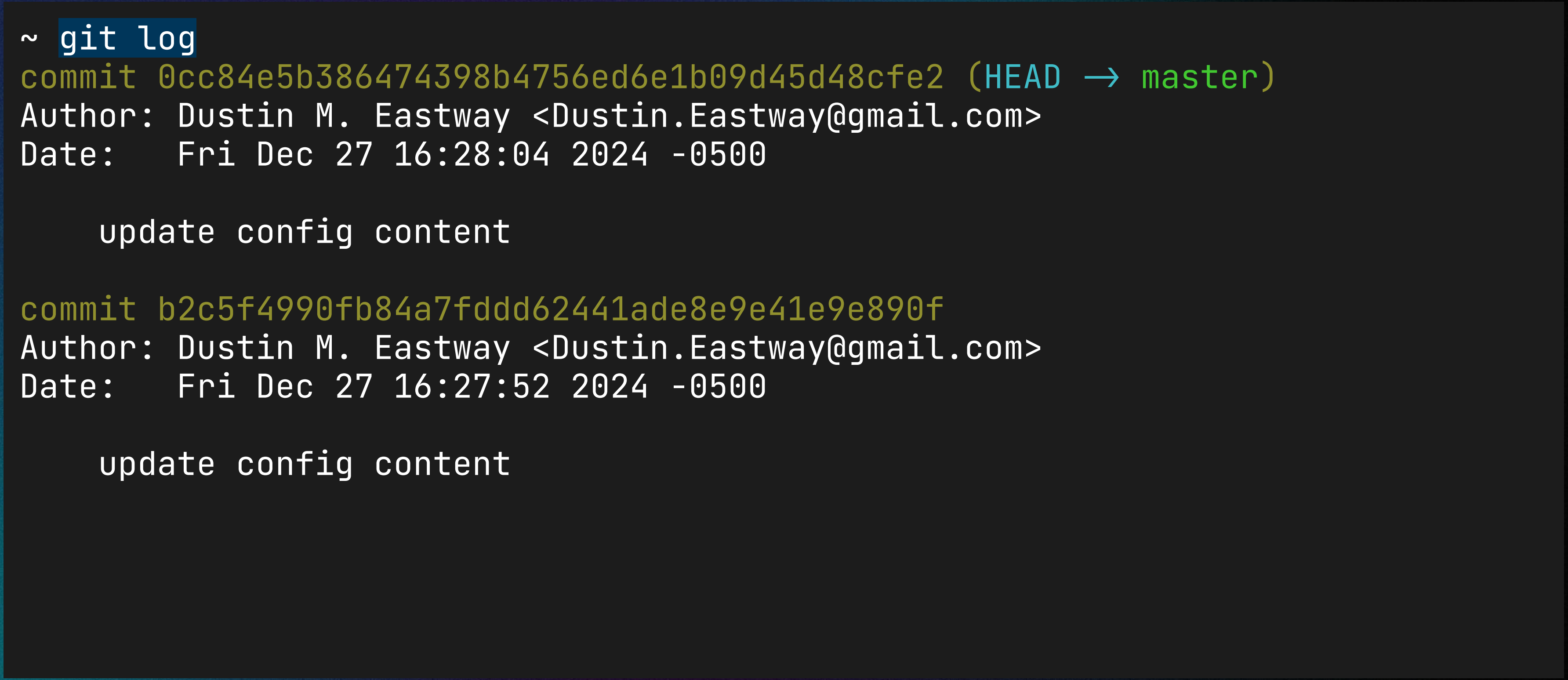
Add commits to display. Continued...



```
~ mv config.txt config.json
~ git add config.json config.txt
~ git commit -m "update config content"
[master 0cc84e5] update config content
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename config.txt => config.json (100%)
~ git status
On branch master
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

How Do I Display Commits?

`git log`



```
~ git log
commit 0cc84e5b386474398b4756ed6e1b09d45d48cfe2 (HEAD -> master)
Author: Dustin M. Eastway <Dustin.Eastway@gmail.com>
Date:   Fri Dec 27 16:28:04 2024 -0500

    update config content

commit b2c5f4990fb84a7fddd62441ade8e9e41e9e890f
Author: Dustin M. Eastway <Dustin.Eastway@gmail.com>
Date:   Fri Dec 27 16:27:52 2024 -0500

    update config content
```

How Do I Display Commits?

Continued...

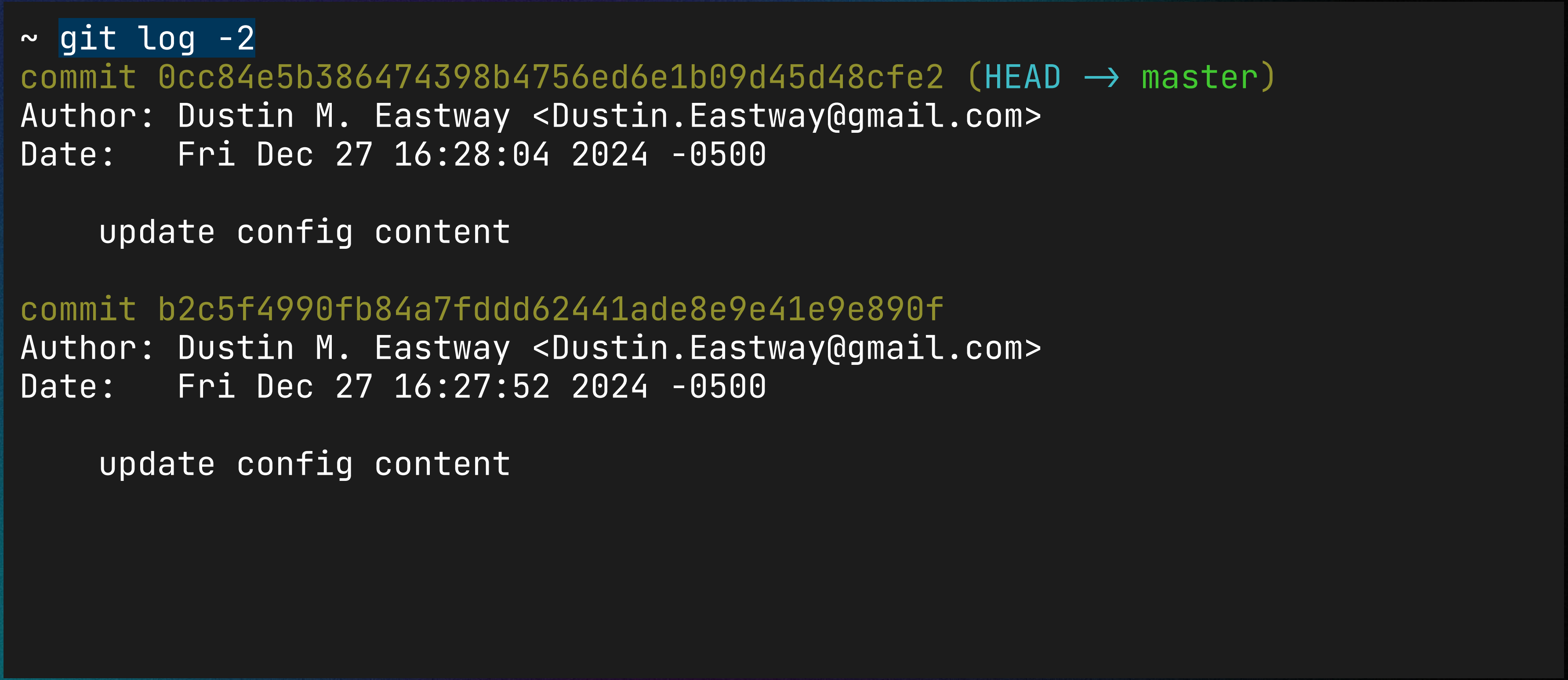
`git log`

```
commit 125a2054d22c0f19b8407bbf647549252ad03428
Author: Dustin M. Eastway <Dustin.Eastway@gmail.com>
Date:   Fri Dec 27 16:20:27 2024 -0500

    test different change types

commit 2d4fc5b5d37fa66c4a1018f55f63ca9ef314e2ad (origin/master)
Author: Dustin M. Eastway <Dustin.Eastway@gmail.com>
Date:   Tue Dec 24 21:44:00 2024 -0500

    initial commit
```

How Do I Display Commits?

Last number of commits.

git log -<max number of commits>

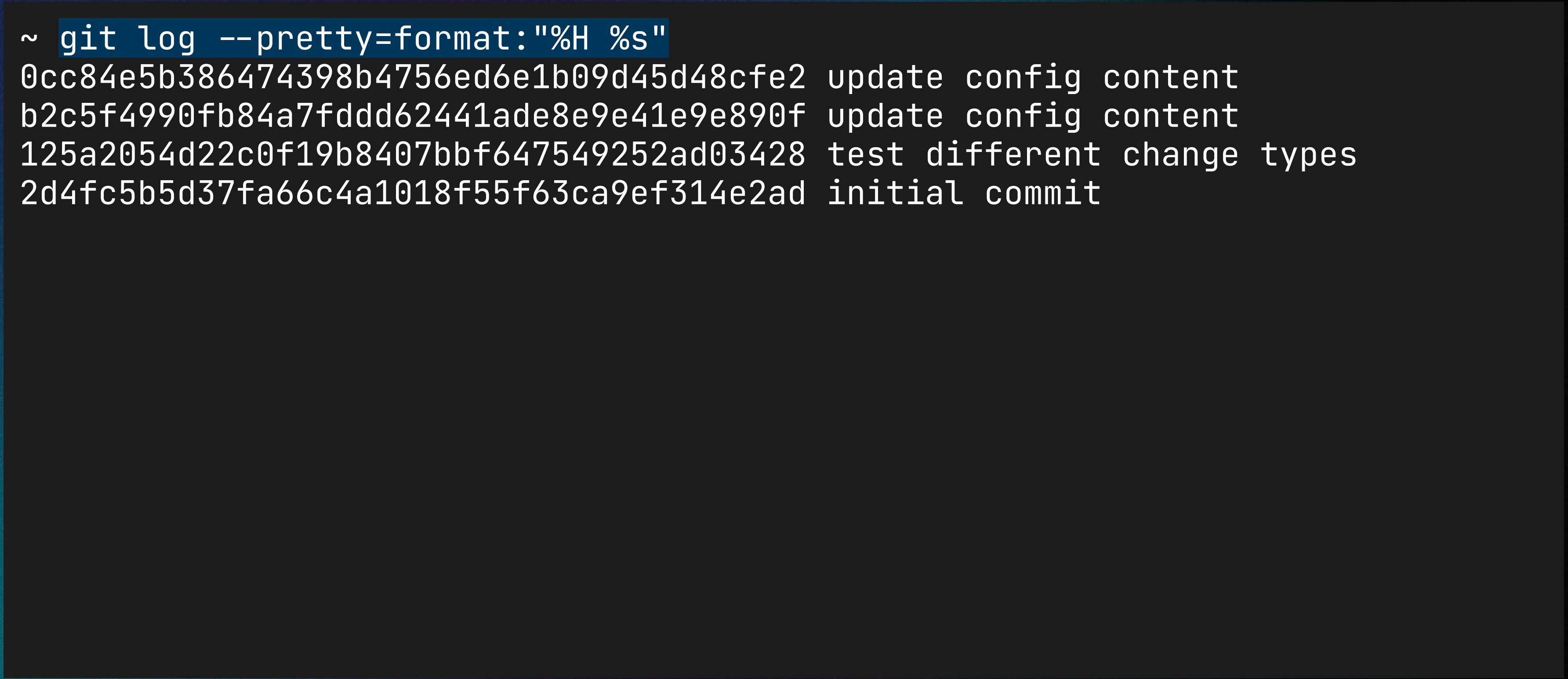


```
~ git log -2
commit 0cc84e5b386474398b4756ed6e1b09d45d48cfe2 (HEAD -> master)
Author: Dustin M. Eastway <Dustin.Eastway@gmail.com>
Date:   Fri Dec 27 16:28:04 2024 -0500

    update config content

commit b2c5f4990fb84a7fddd62441ade8e9e41e9e890f
Author: Dustin M. Eastway <Dustin.Eastway@gmail.com>
Date:   Fri Dec 27 16:27:52 2024 -0500

    update config content
```

How Do I Display Commits?

With formatting.

`git log --pretty=format:"<log format>"`

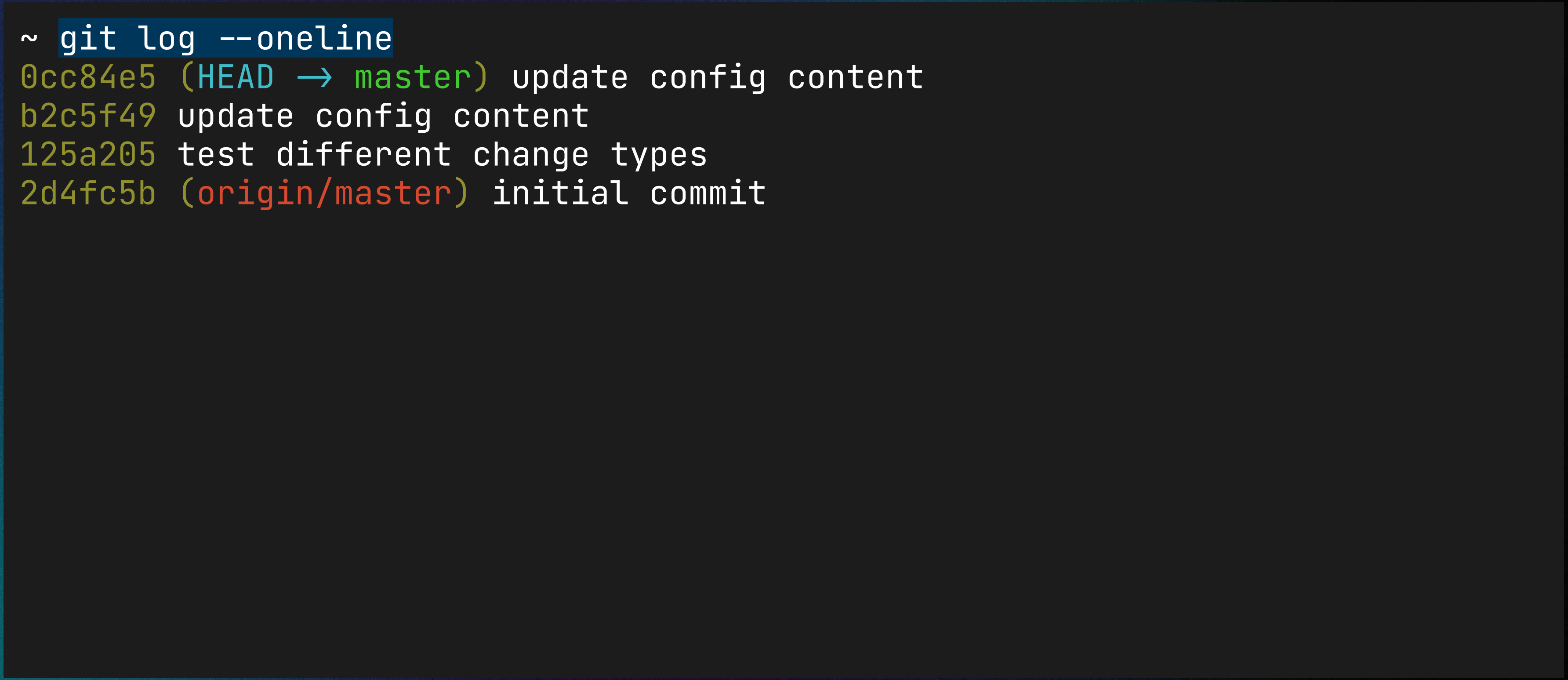


```
~ git log --pretty=format:"%H %s"
0cc84e5b386474398b4756ed6e1b09d45d48cfe2 update config content
b2c5f4990fb84a7fddd62441ade8e9e41e9e890f update config content
125a2054d22c0f19b8407bbf647549252ad03428 test different change types
2d4fc5b5d37fa66c4a1018f55f63ca9ef314e2ad initial commit
```

How Do I Display Commits?

On single lines.

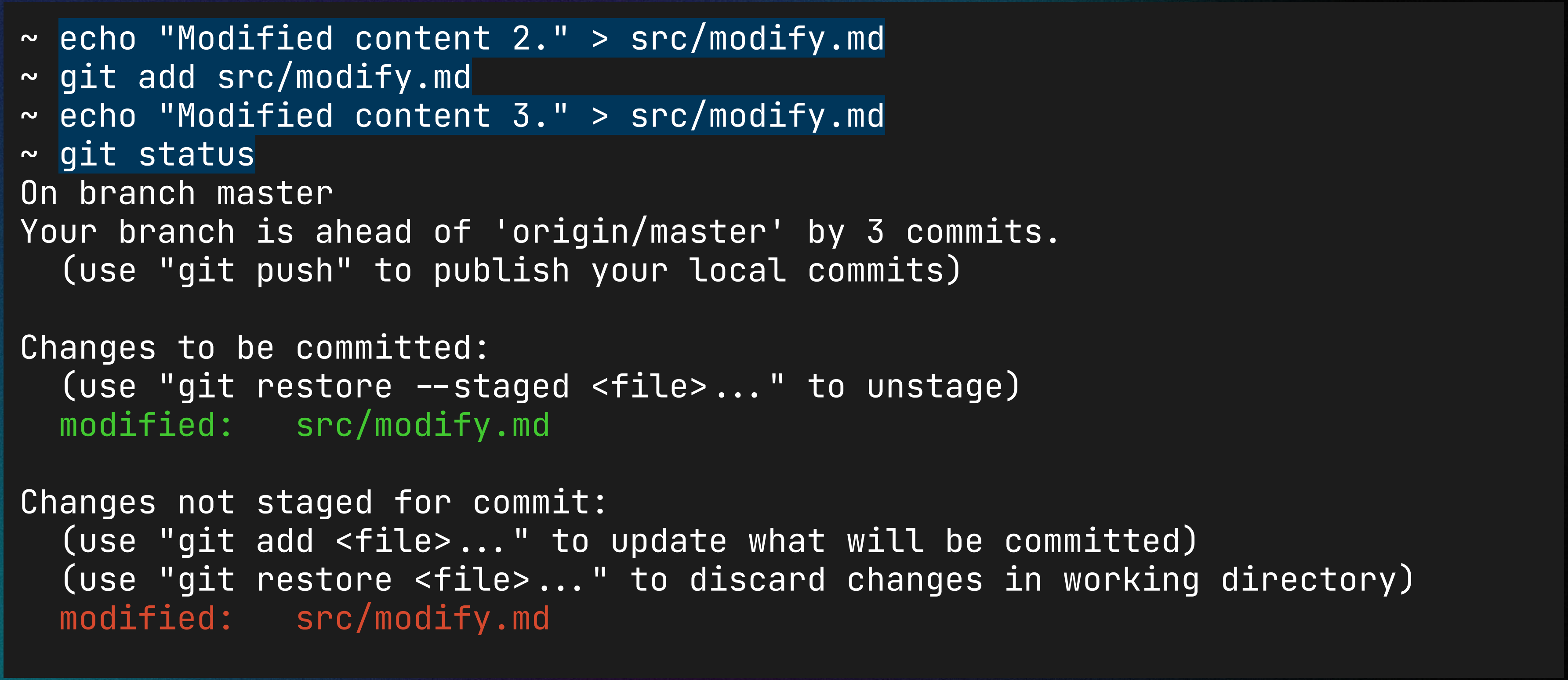
git log --oneline



```
~ git log --oneline
0cc84e5 (HEAD -> master) update config content
b2c5f49 update config content
125a205 test different change types
2d4fc5b (origin/master) initial commit
```

Prep Content

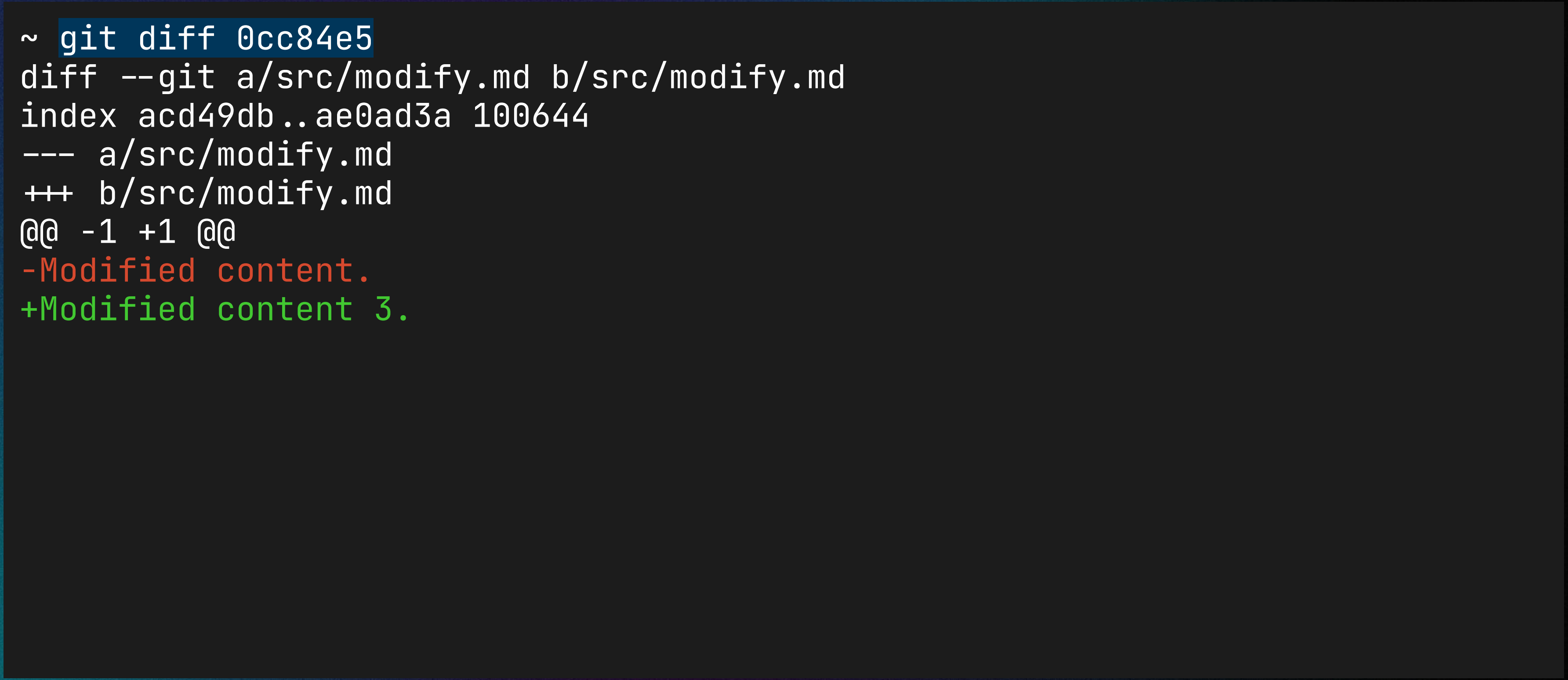
Add staged & unstaged changes.



```
~ echo "Modified content 2." > src/modify.md
~ git add src/modify.md
~ echo "Modified content 3." > src/modify.md
~ git status
On branch master
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  modified:   src/modify.md

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
  modified:   src/modify.md
```

How Do I Display Changes?

For all changes after <commit hash>.
git diff <commit hash>



```
~ git diff 0cc84e5
diff --git a/src/modify.md b/src/modify.md
index acd49db..ae0ad3a 100644
--- a/src/modify.md
+++ b/src/modify.md
@@ -1 +1 @@
-Modified content.
+Modified content 3.
```

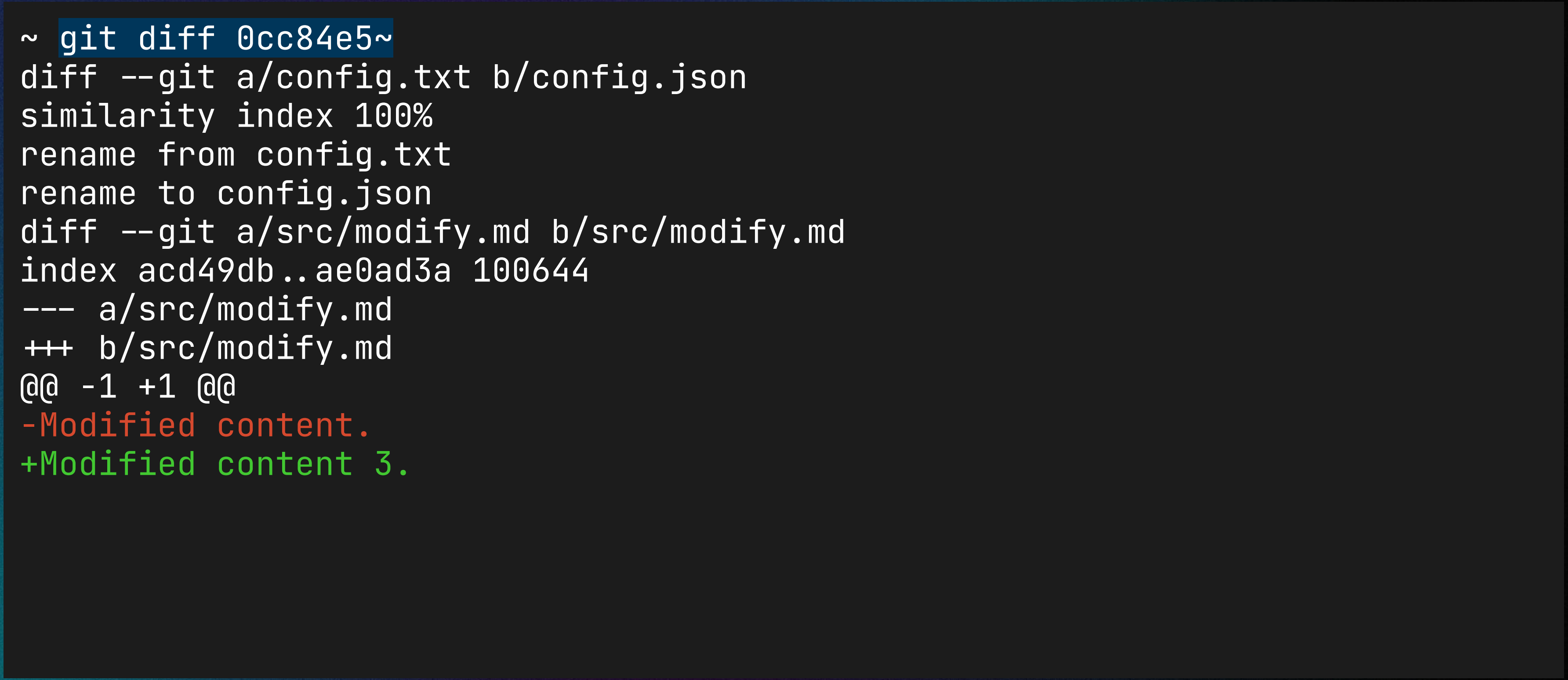
Note: Commit Hash Arguments

You can offset commit hashes using `~`.

```
Given that these are your last commit hashes (they will not be).
0cc84e5 (HEAD → master) update config content
b2c5f49 update config content
125a205 test different change types
2d4fc5b (origin/master) initial commit

Then the following is true.
 ●  0cc84e5 is the latest commit.
 ●  0cc84e5~1 is the commit before 0cc84e5 (b2c5f49).
 ●  0cc84e5~2 is 2 commits before 0cc84e5 (125a205).
 ●  0cc84e5~ is a shorthand for 0cc84e5~1 (b2c5f49).
```

How Do I Display Changes?

For all changes in and after <commit hash>.
git diff <commit hash>~

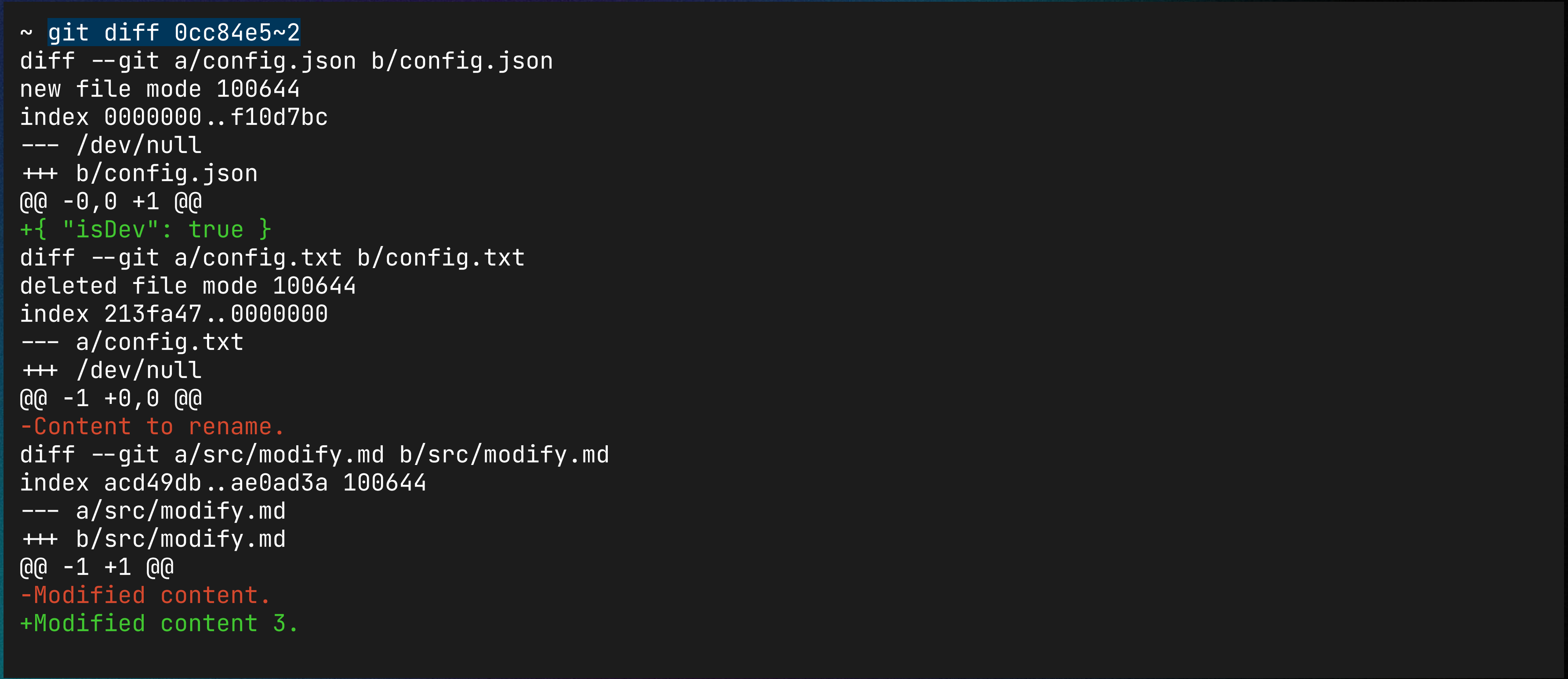


```
~ git diff 0cc84e5~
diff --git a/config.txt b/config.json
similarity index 100%
rename from config.txt
rename to config.json
diff --git a/src/modify.md b/src/modify.md
index acd49db..ae0ad3a 100644
--- a/src/modify.md
+++ b/src/modify.md
@@ -1 +1 @@
-Modified content.
+Modified content 3.
```

How Do I Display Changes?

For all changes after <number> commits before <commit hash>.

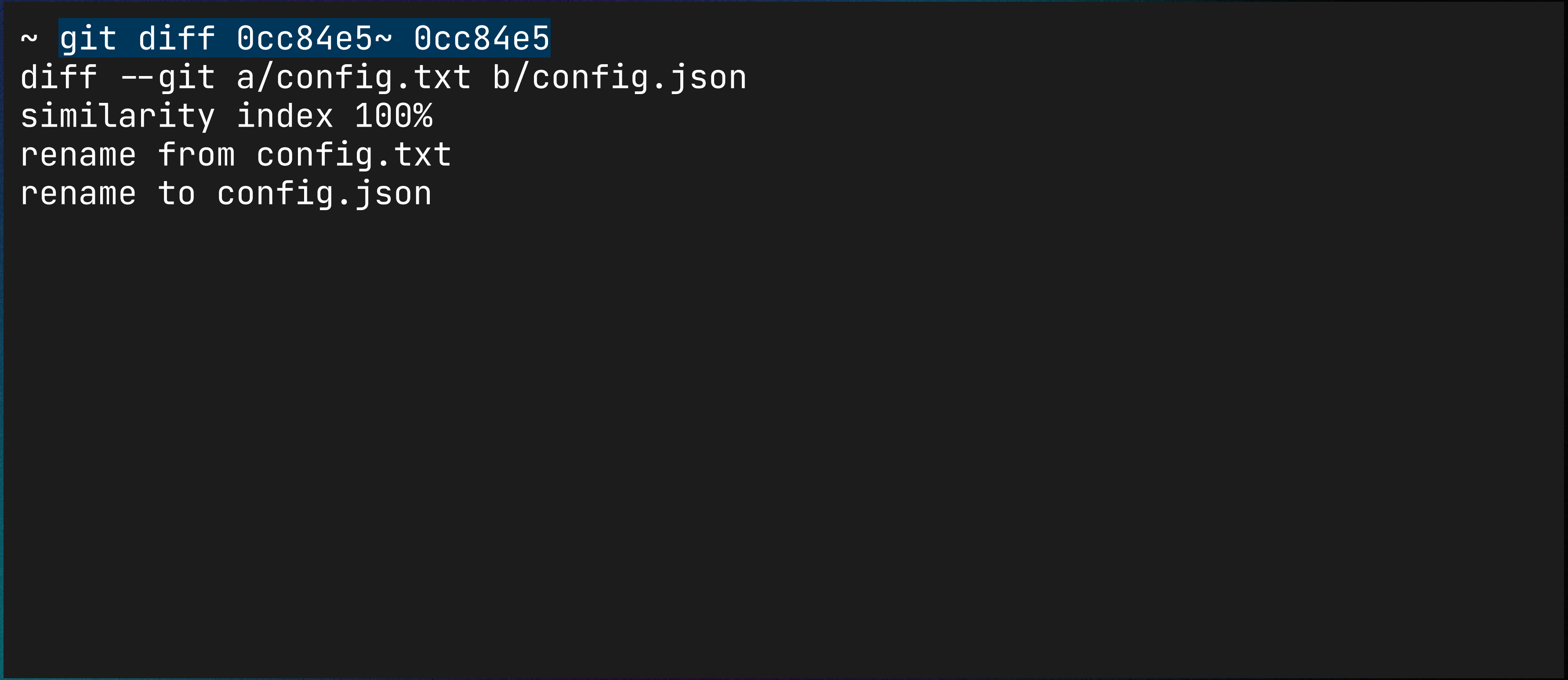
git diff <commit hash>~<number>



```
~ git diff 0cc84e5~2
diff --git a/config.json b/config.json
new file mode 100644
index 0000000..f10d7bc
--- /dev/null
+++ b/config.json
@@ -0,0 +1 @@
+{ "isDev": true }
diff --git a/config.txt b/config.txt
deleted file mode 100644
index 213fa47..0000000
--- a/config.txt
+++ /dev/null
@@ -1 +0,0 @@
-Content to rename.
diff --git a/src/modify.md b/src/modify.md
index acd49db..ae0ad3a 100644
--- a/src/modify.md
+++ b/src/modify.md
@@ -1 +1 @@
-Modified content.
+Modified content 3.
```

How Do I Display Changes?

For all changes in <commit hash>.
git diff <commit hash>~ <commit hash>

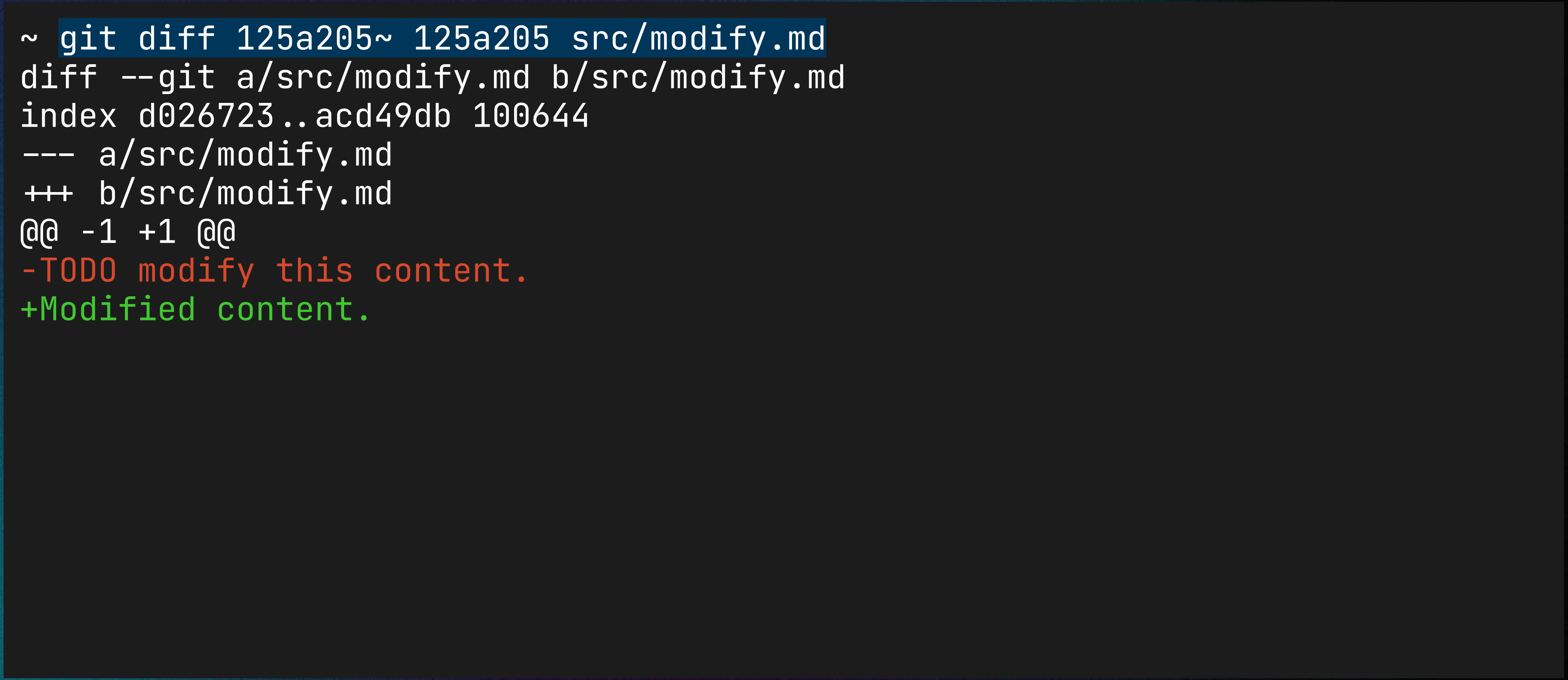


```
~ git diff 0cc84e5~ 0cc84e5
diff --git a/config.txt b/config.json
similarity index 100%
rename from config.txt
rename to config.json
```

How Do I Display Changes?

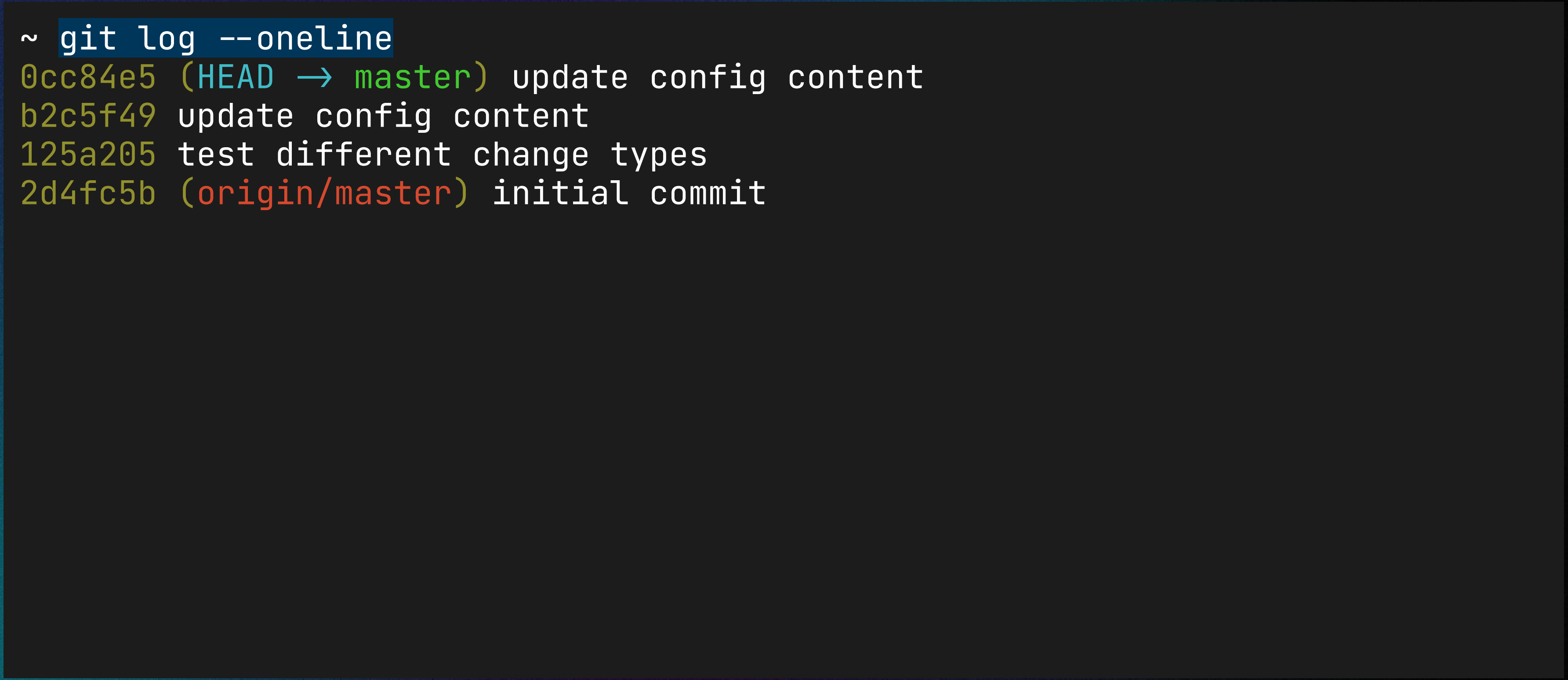
For all changes in <commit hash> in specific file(s).

git diff <commit hash>~ <commit hash> <path to file(s)>



```
~ git diff 125a205~ 125a205 src/modify.md
diff --git a/src/modify.md b/src/modify.md
index d026723..acd49db 100644
--- a/src/modify.md
+++ b/src/modify.md
@@ -1 +1 @@
-TODO modify this content.
+Modified content.
```

Term: HEAD

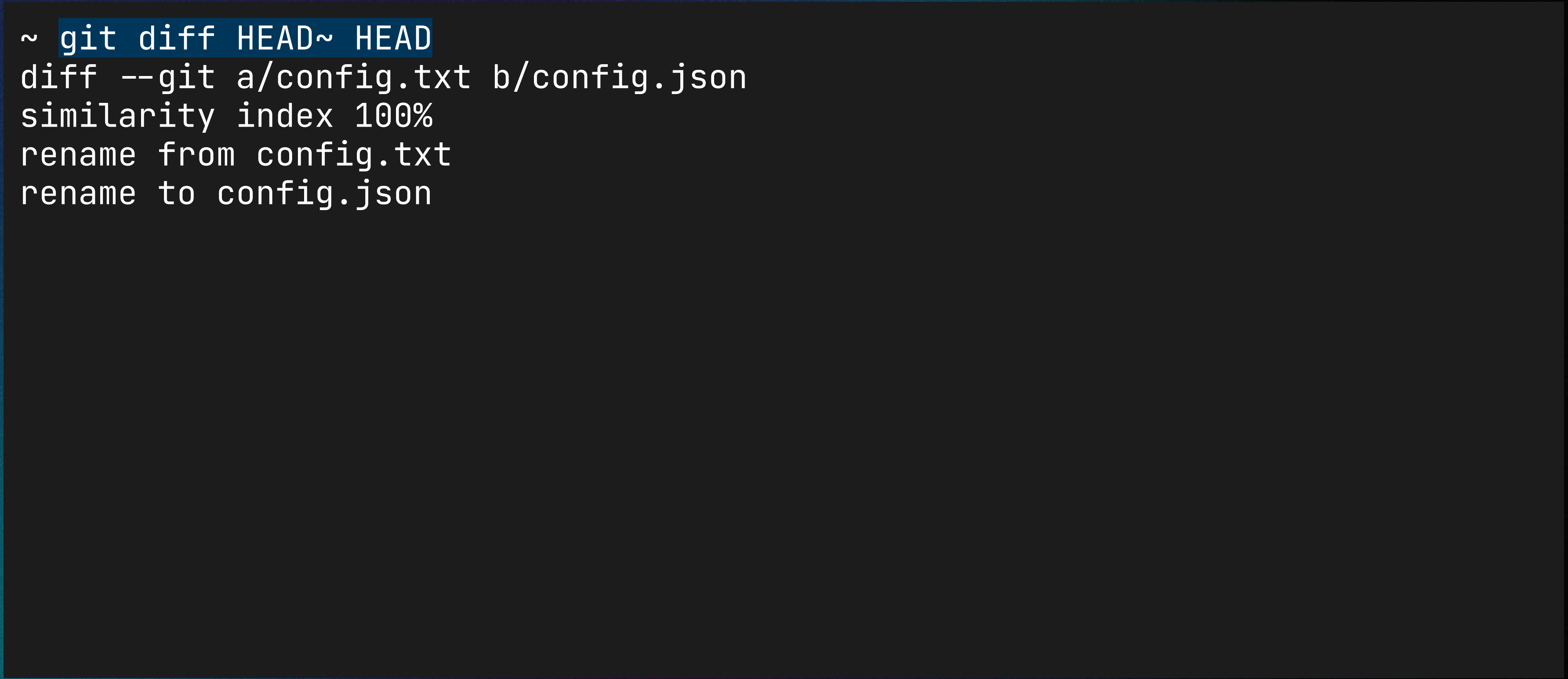


```
~ git log --oneline
0cc84e5 (HEAD → master) update config content
b2c5f49 update config content
125a205 test different change types
2d4fc5b (origin/master) initial commit
```

How Do I Display Changes?

For all changes in the last commit.

`git diff HEAD~ HEAD`

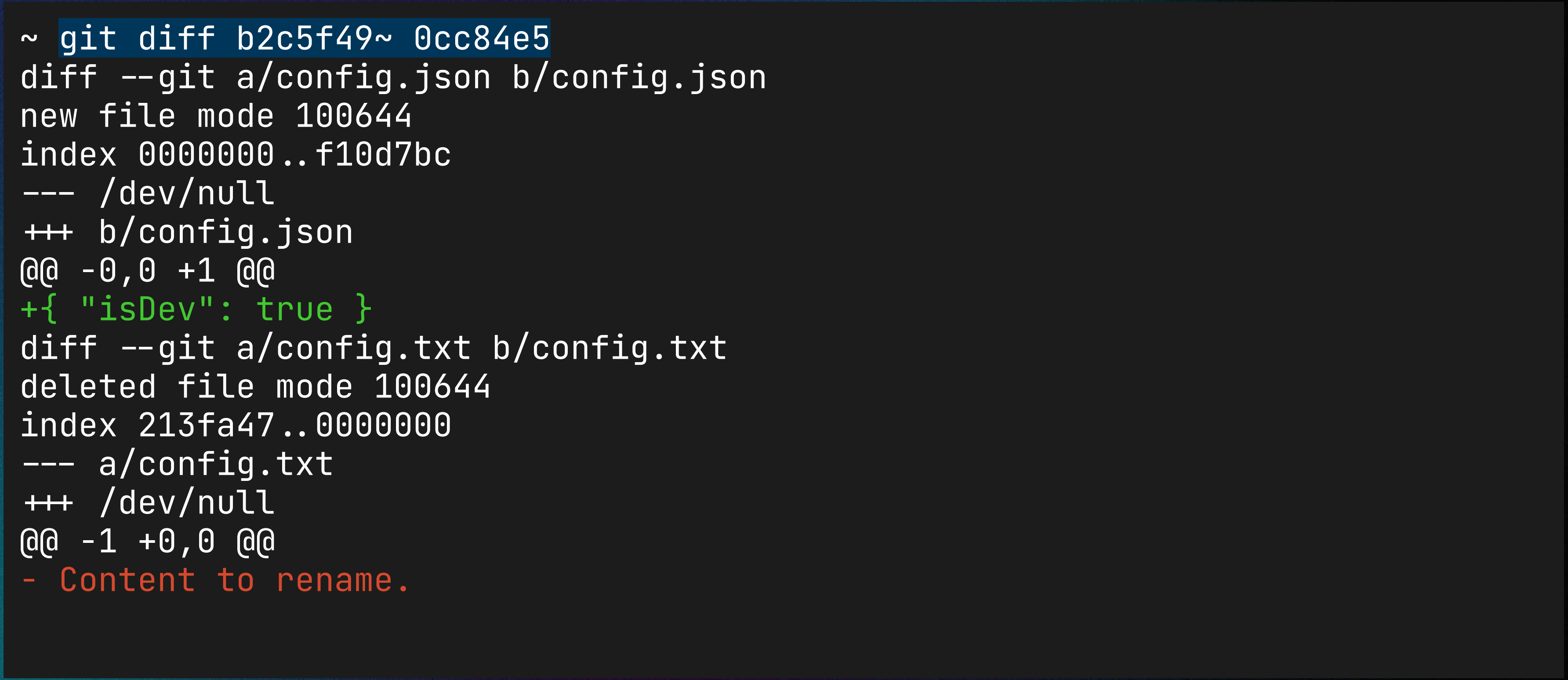


```
~ git diff HEAD~ HEAD
diff --git a/config.txt b/config.json
similarity index 100%
rename from config.txt
rename to config.json
```

How Do I Display Changes?

For all changes from <start commit hash> to <end commit hash> (inclusive).

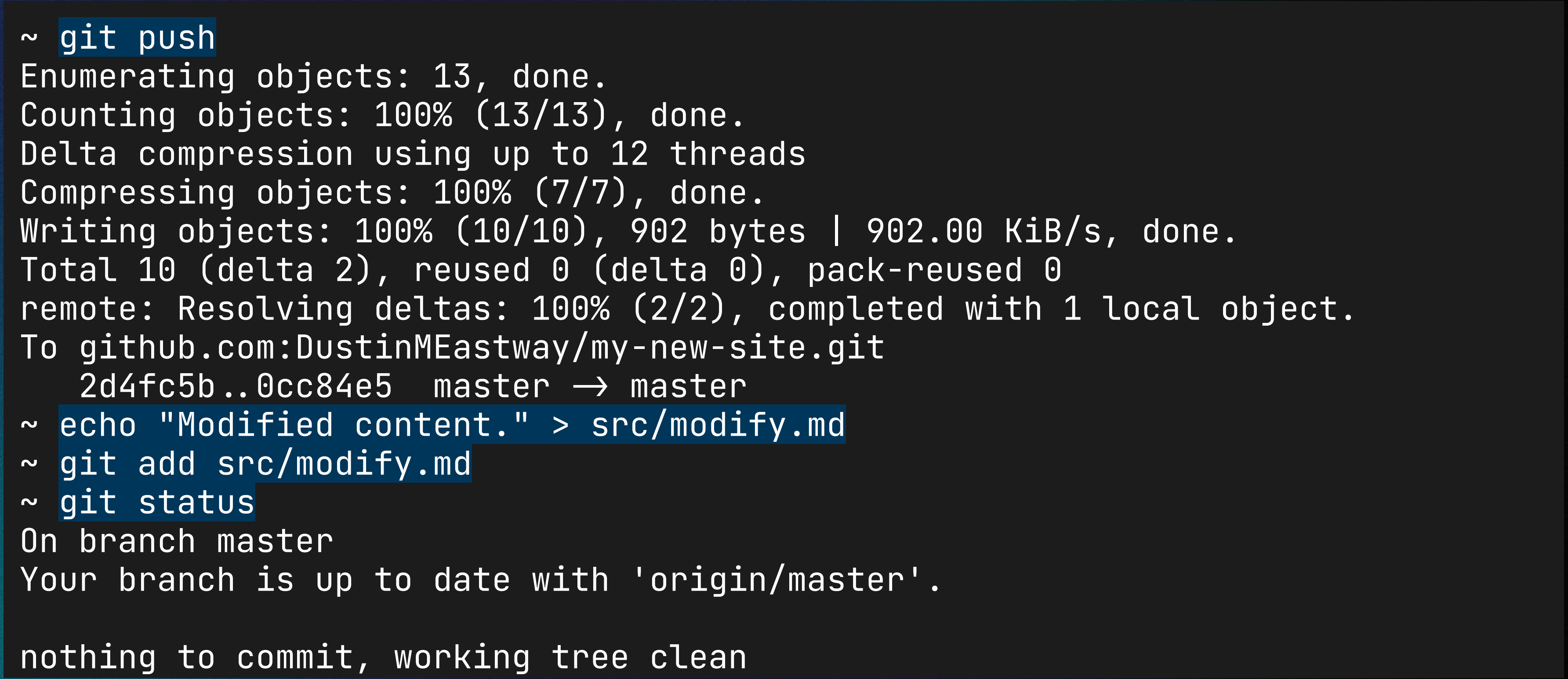
git diff <start commit hash>~ <end commit hash>



```
~ git diff b2c5f49~ 0cc84e5
diff --git a/config.json b/config.json
new file mode 100644
index 0000000..f10d7bc
--- /dev/null
+++ b/config.json
@@ -0,0 +1 @@
+{ "isDev": true }
diff --git a/config.txt b/config.txt
deleted file mode 100644
index 213fa47..0000000
--- a/config.txt
+++ /dev/null
@@ -1 +0,0 @@
- Content to rename.
```

Prep Content

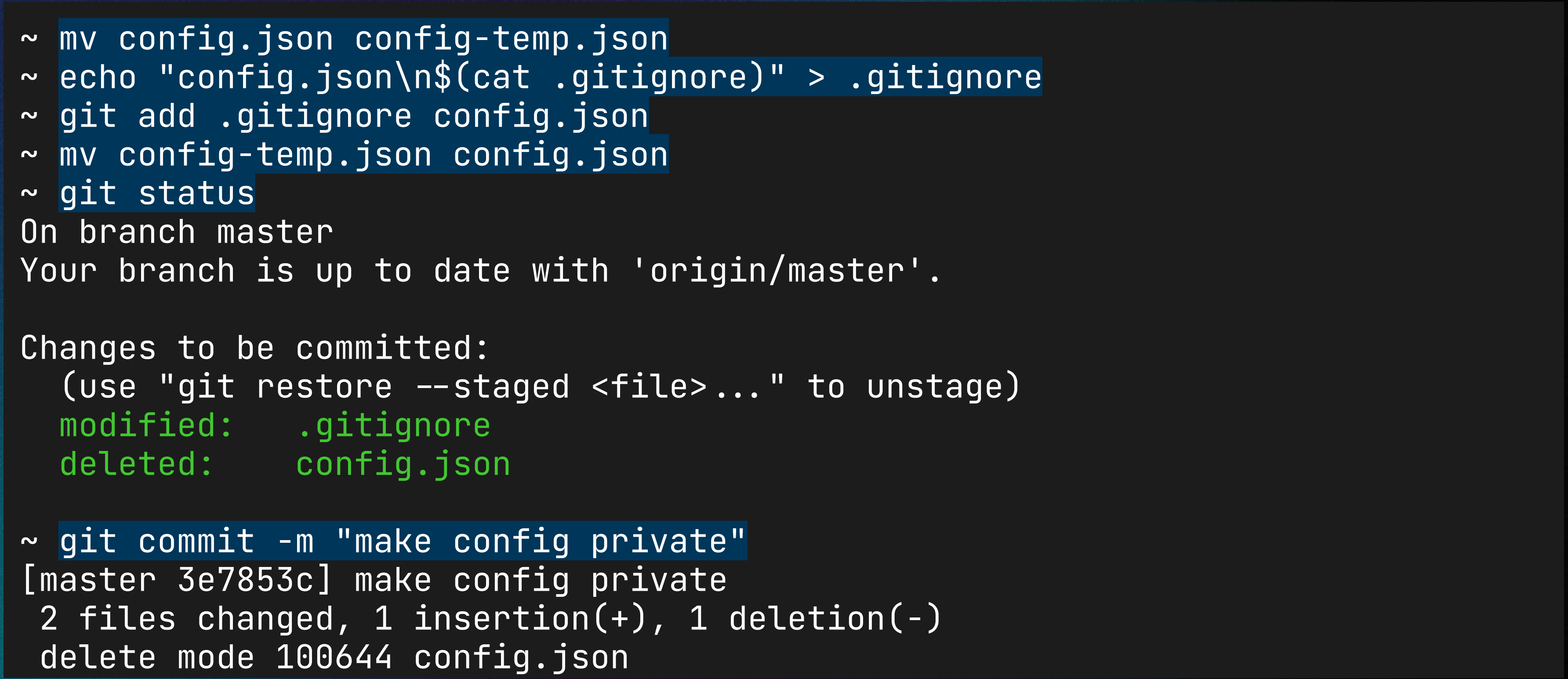
Ready changes to be undone.



```
~ git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 902 bytes | 902.00 KiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:DustinMEastway/my-new-site.git
   2d4fc5b..0cc84e5  master -> master
~ echo "Modified content." > src/modify.md
~ git add src/modify.md
~ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

How Do I Ignore Files?

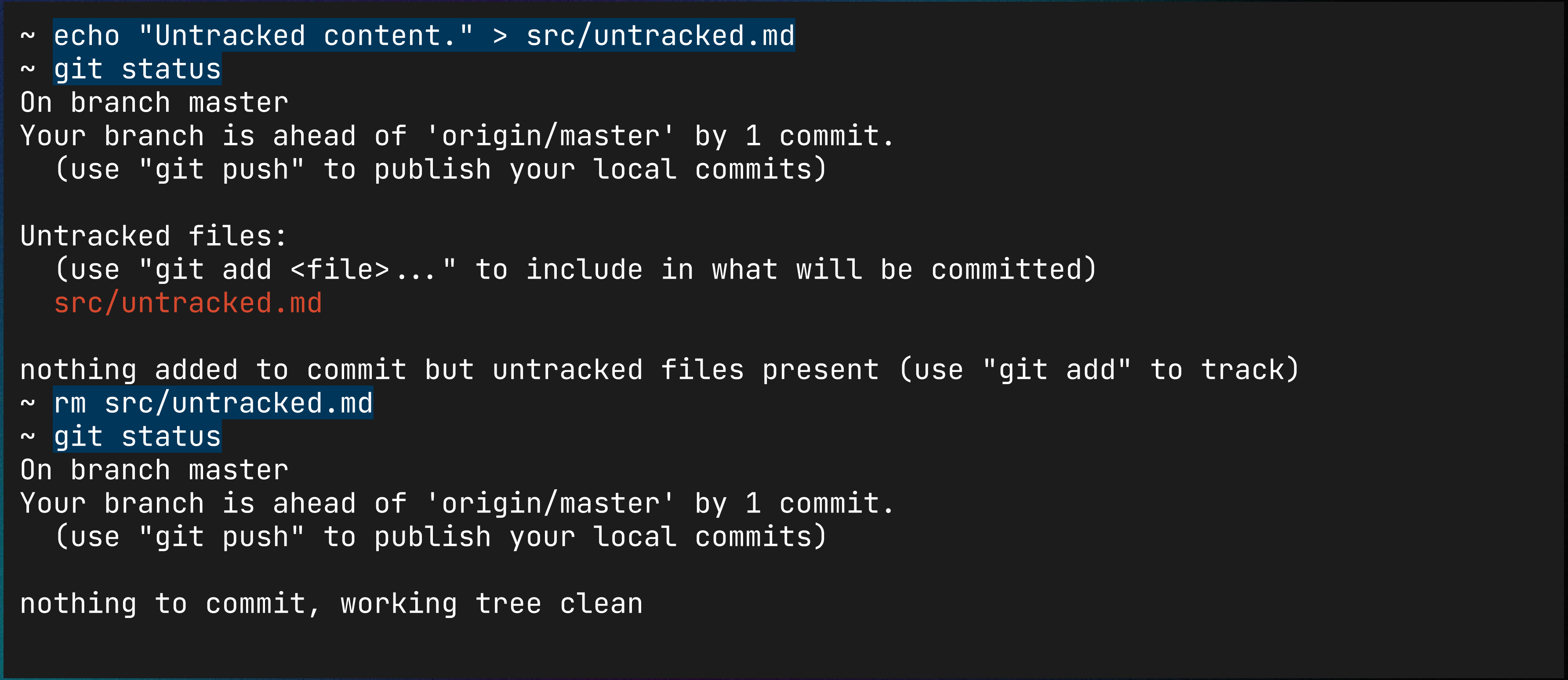
Modified, staged, & committed file(s).
Add .gitignore patterns.



```
~ mv config.json config-temp.json
~ echo "config.json\n$(cat .gitignore)" > .gitignore
~ git add .gitignore config.json
~ mv config-temp.json config.json
~ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  modified:   .gitignore
  deleted:    config.json

~ git commit -m "make config private"
[master 3e7853c] make config private
 2 files changed, 1 insertion(+), 1 deletion(-)
 delete mode 100644 config.json
```

How Do I Undo Changes?

Untracked changes.



```
~ echo "Untracked content." > src/untracked.md
~ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  src/untracked.md

nothing added to commit but untracked files present (use "git add" to track)
~ rm src/untracked.md
~ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

How Do I Undo Changes?

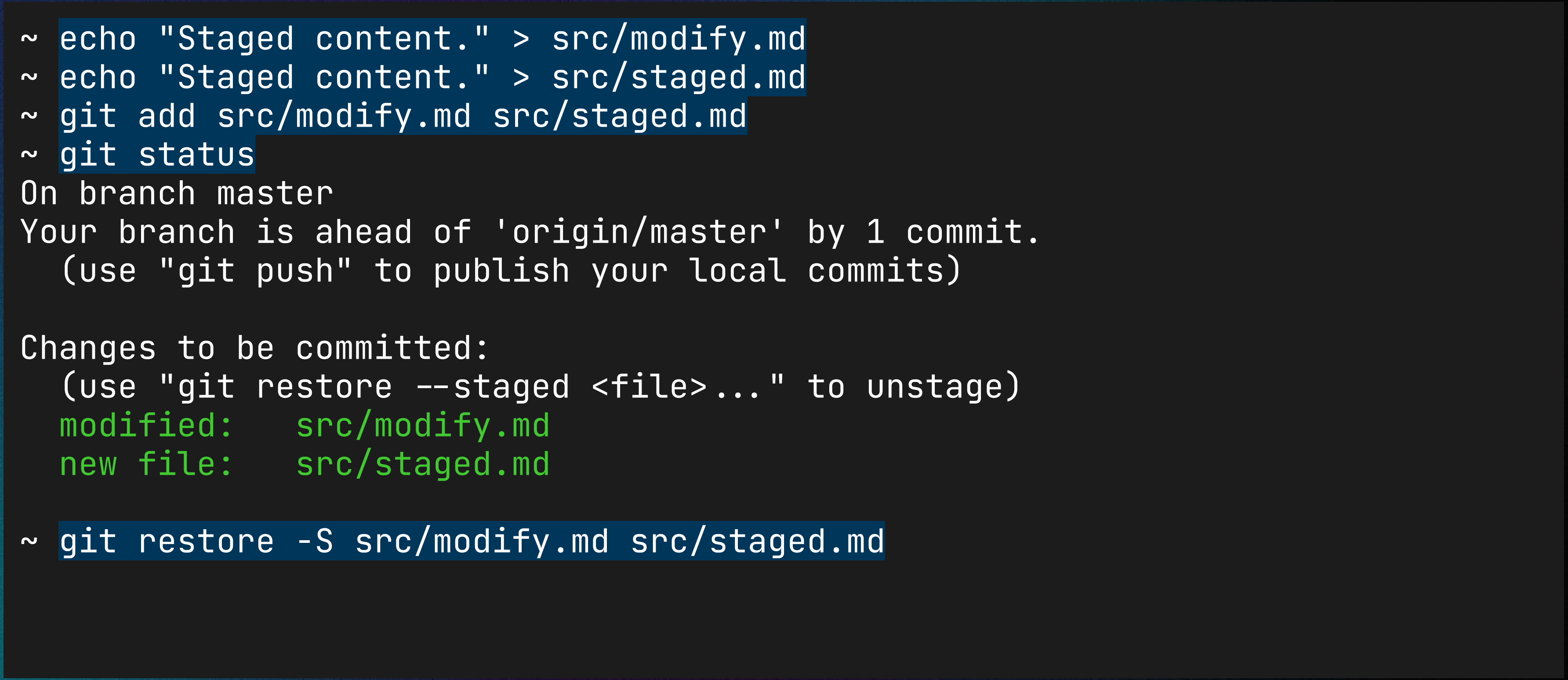
Unstaged changes.

git restore [-W] <path to file(s)>

```
~ echo "Unstaged content." > src/modify.md
~ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
  modified:   src/modify.md

no changes added to commit (use "git add" and/or "git commit -a")
~ git restore src/modify.md
~ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

How Do I Unstage Staged Changes?

`git restore -S <path to file(s)>`

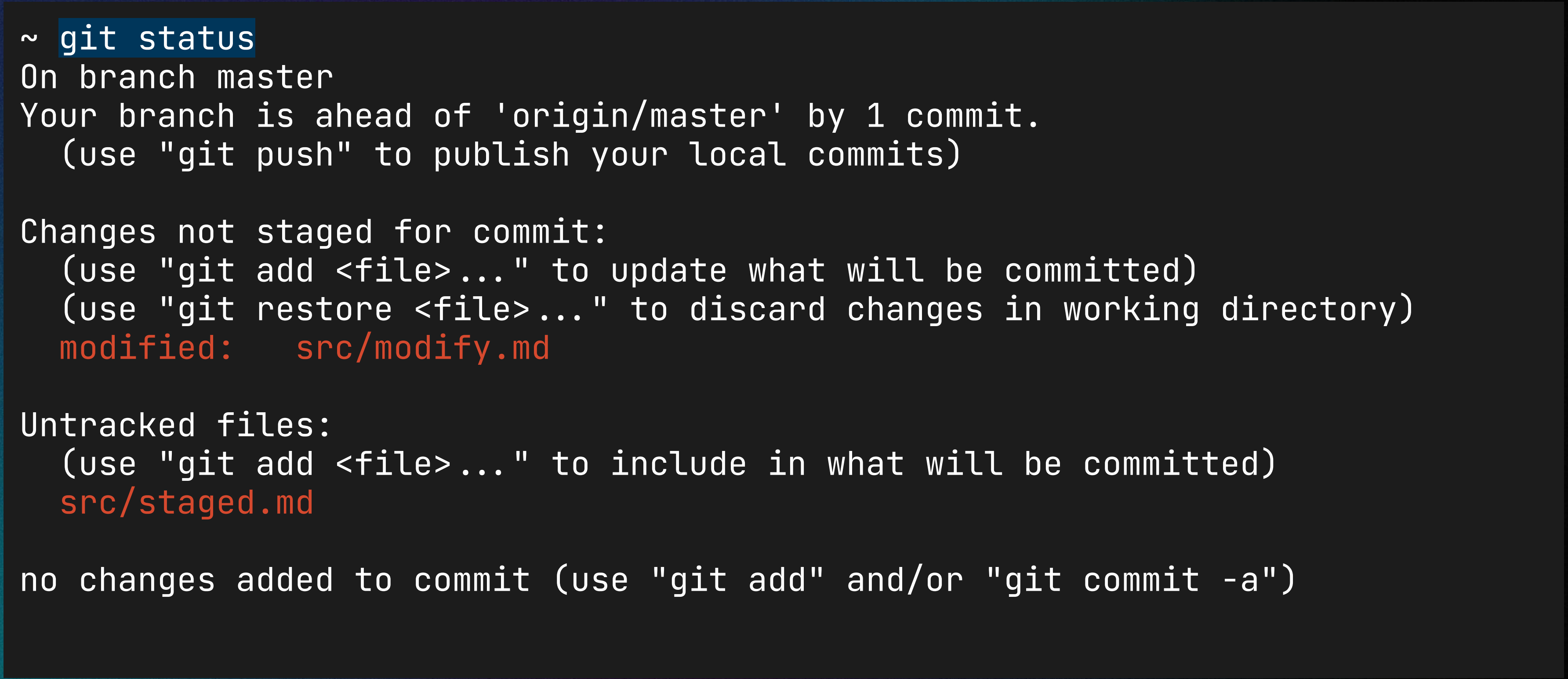


```
~ echo "Staged content." > src/modify.md
~ echo "Staged content." > src/staged.md
~ git add src/modify.md src/staged.md
~ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
	modified:   src/modify.md
	new file:   src/staged.md

~ git restore -S src/modify.md src/staged.md
```

How Do I Unstage Staged Changes?

Continued...

git restore -S <path to file(s)>

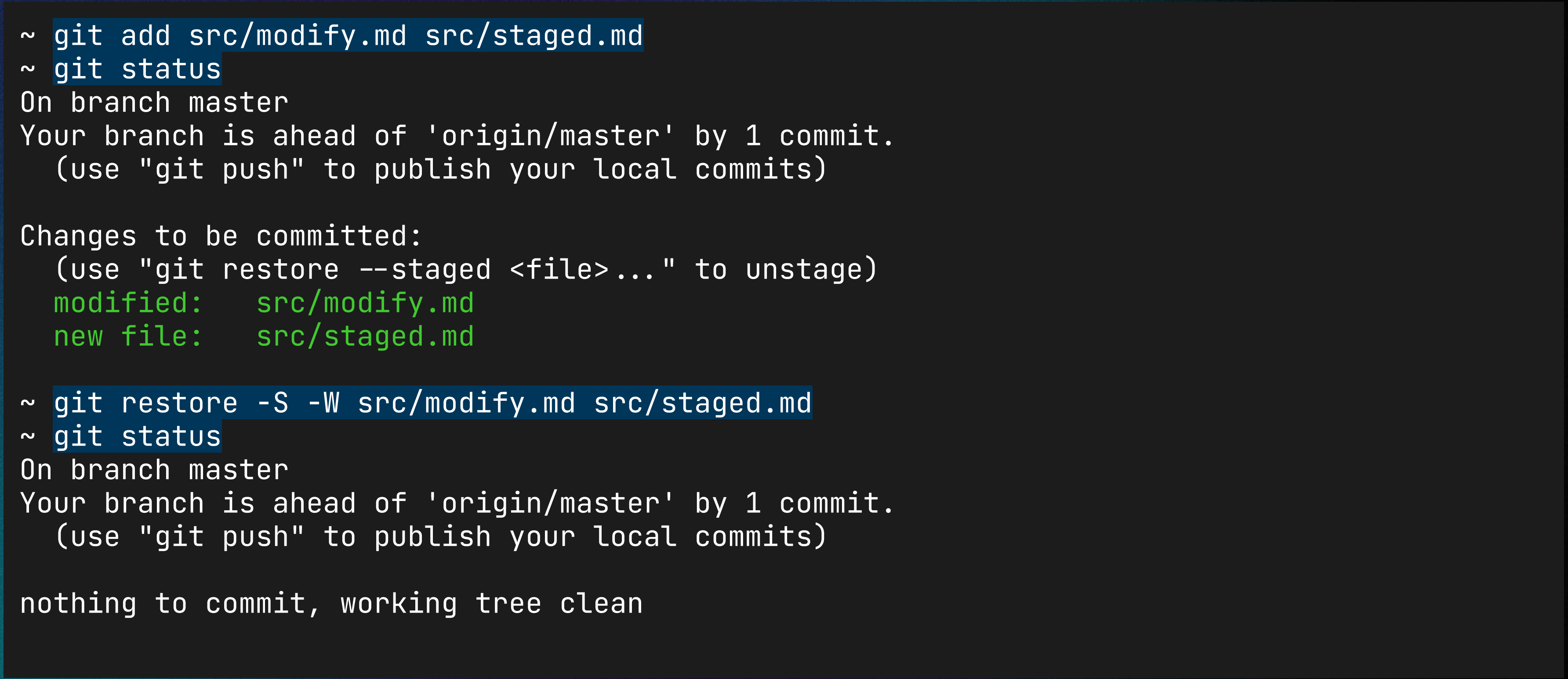


```
~ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
  modified:   src/modify.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  src/staged.md

no changes added to commit (use "git add" and/or "git commit -a")
```

How Do I Undo Changes?

Staged changes.

git restore -S -W <path to file(s)>

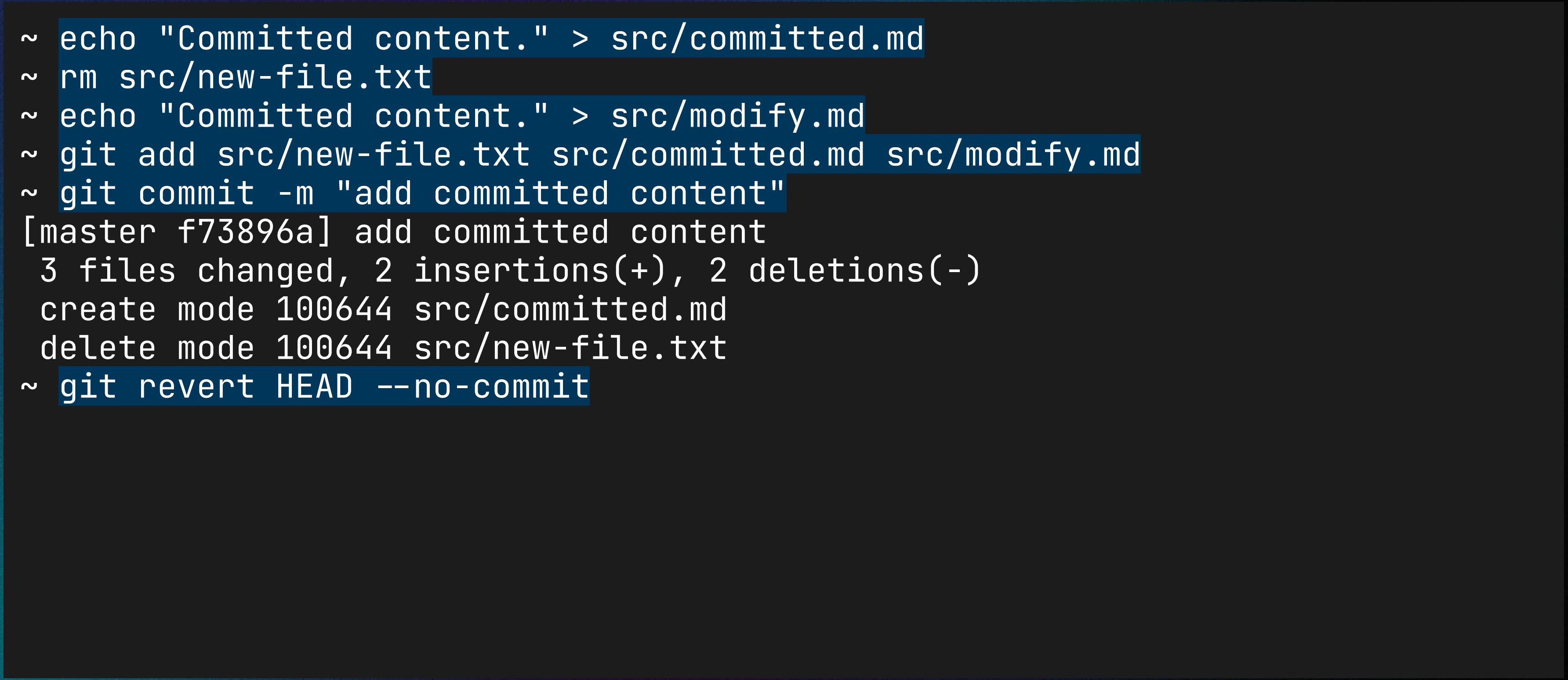


```
~ git add src/modify.md src/staged.md
~ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  modified:   src/modify.md
  new file:   src/staged.md

~ git restore -S -W src/modify.md src/staged.md
~ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

How Do I Undo Changes?

Revert committed changes.

`git revert <commit hash> [--no-commit]`

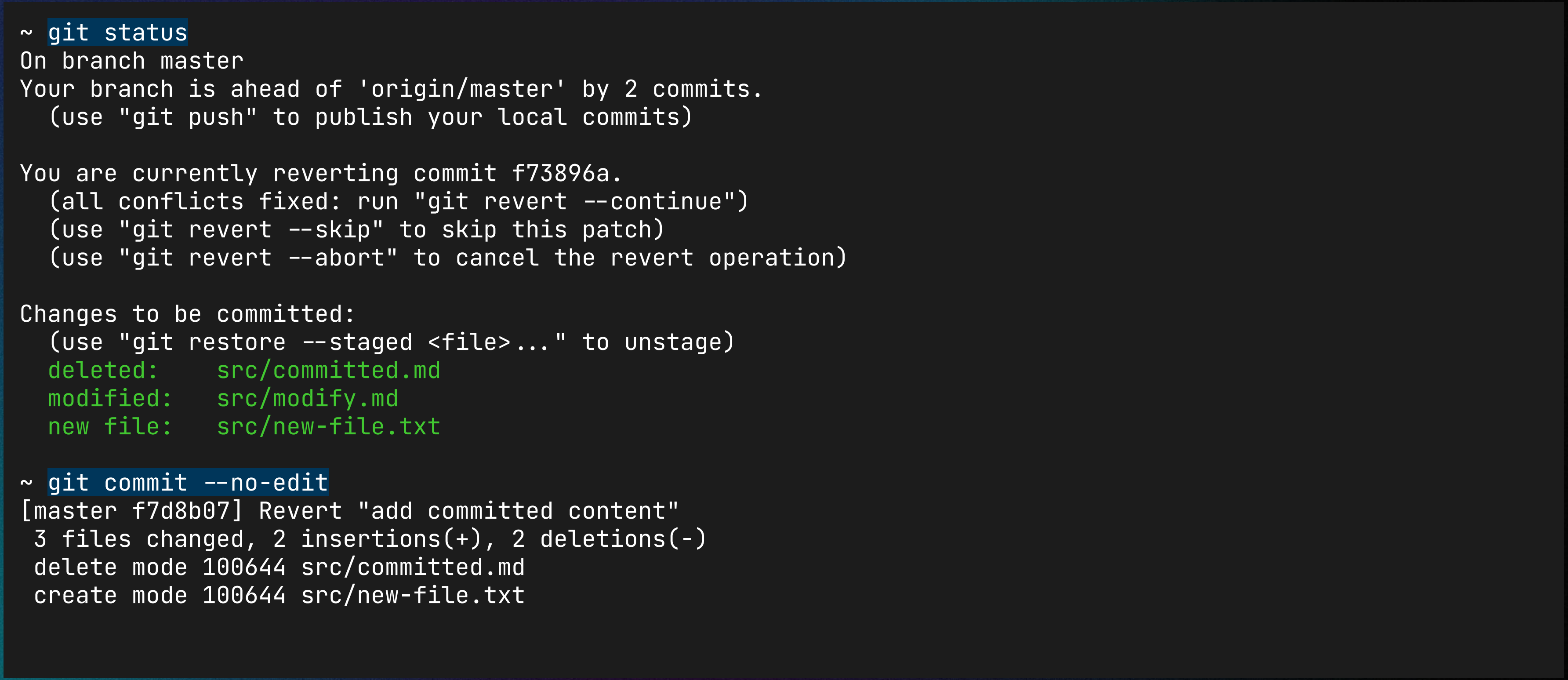


```
~ echo "Committed content." > src/committed.md
~ rm src/new-file.txt
~ echo "Committed content." > src/modify.md
~ git add src/new-file.txt src/committed.md src/modify.md
~ git commit -m "add committed content"
[master f73896a] add committed content
 3 files changed, 2 insertions(+), 2 deletions(-)
 create mode 100644 src/committed.md
 delete mode 100644 src/new-file.txt
~ git revert HEAD --no-commit
```

How Do I Undo Changes?

Revert committed changes. Continued...

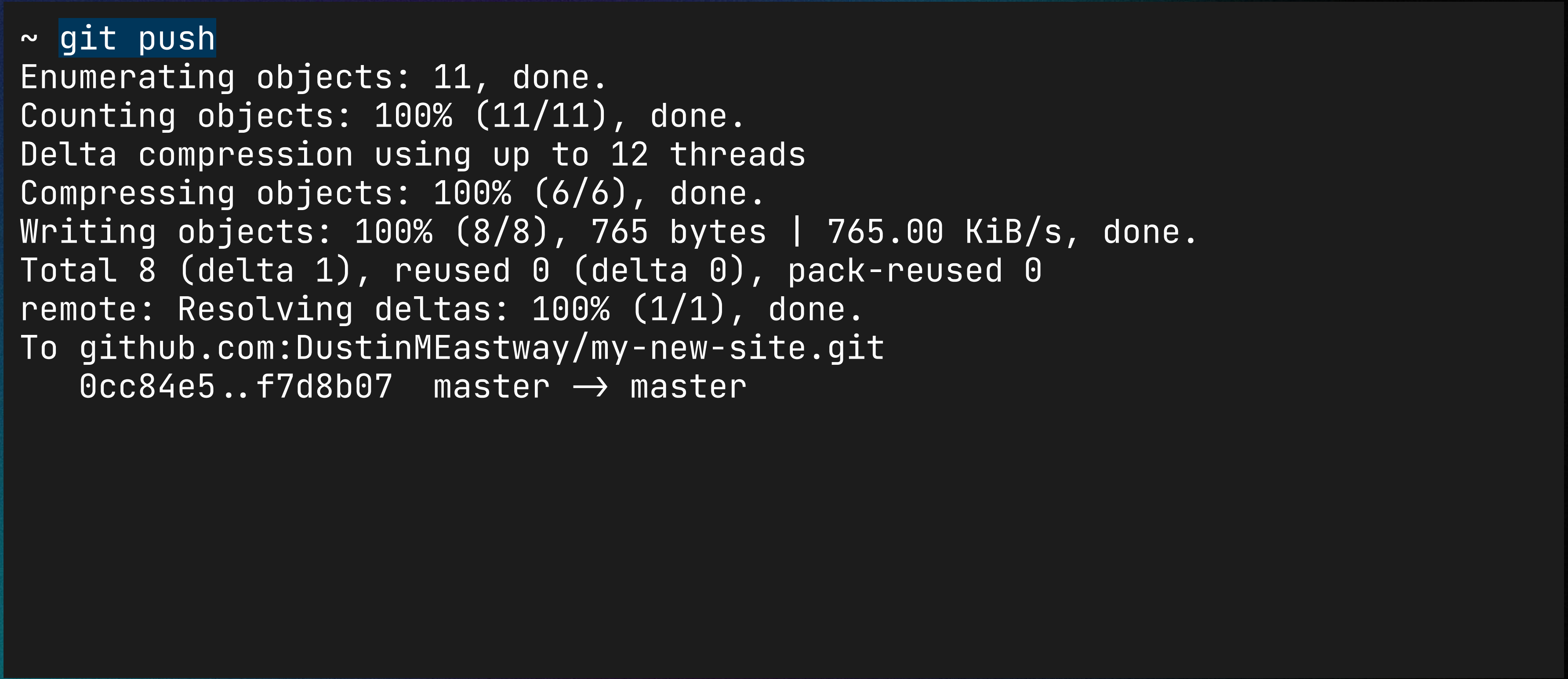
git revert <commit hash> [--no-commit]



```
~ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

You are currently reverting commit f73896a.
  (all conflicts fixed: run "git revert --continue")
  (use "git revert --skip" to skip this patch)
  (use "git revert --abort" to cancel the revert operation)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  deleted:    src/committed.md
  modified:   src/modify.md
  new file:   src/new-file.txt

~ git commit --no-edit
[master f7d8b07] Revert "add committed content"
 3 files changed, 2 insertions(+), 2 deletions(-)
 delete mode 100644 src/committed.md
 create mode 100644 src/new-file.txt
```

Prep Content

Ready more changes to be undone.

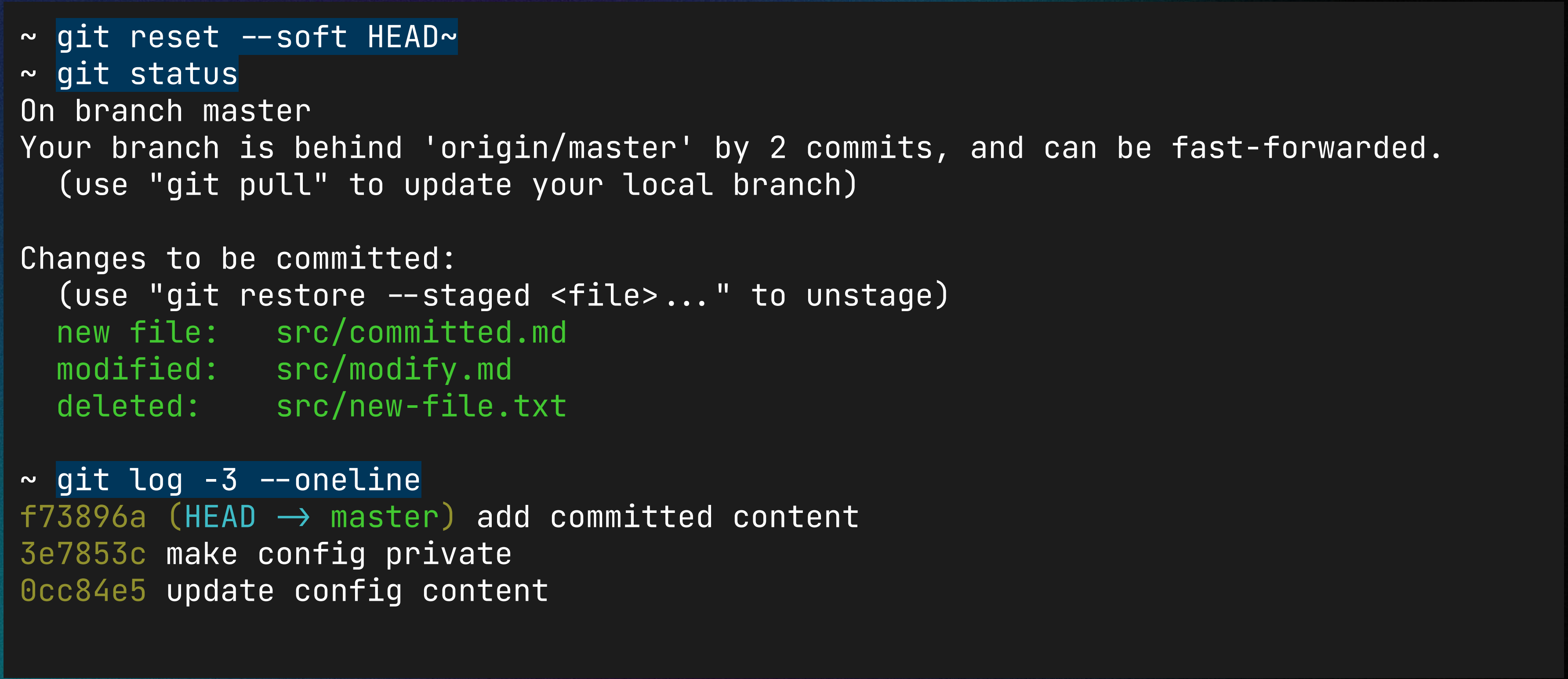


```
~ git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 765 bytes | 765.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:DustinMEastway/my-new-site.git
   0cc84e5..f7d8b07  master -> master
```

How Do I Undo Changes?

Remove commit while keeping changes.

`git reset --soft <commit hash to move to>`

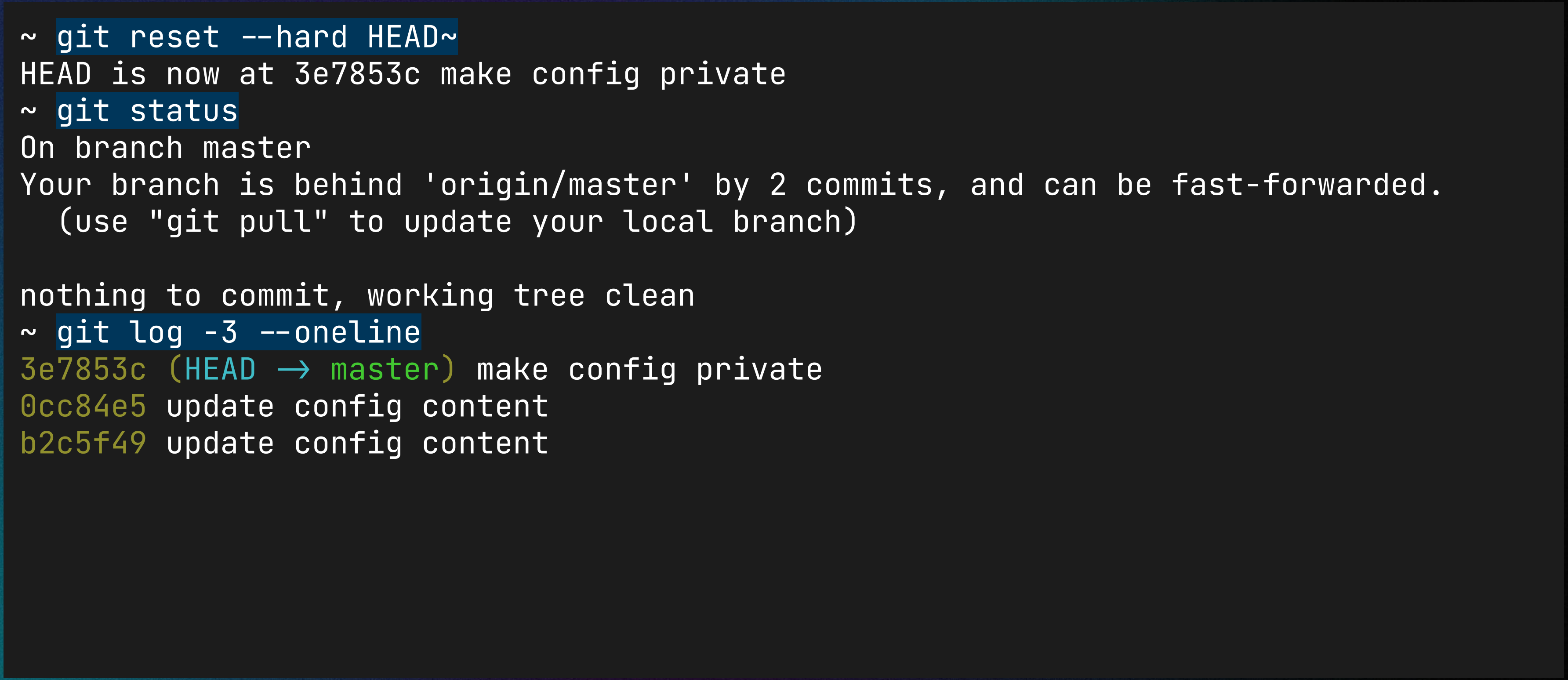


```
~ git reset --soft HEAD~
~ git status
On branch master
Your branch is behind 'origin/master' by 2 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  new file:   src/committed.md
  modified:   src/modify.md
  deleted:    src/new-file.txt

~ git log -3 --oneline
f73896a (HEAD → master) add committed content
3e7853c make config private
0cc84e5 update config content
```

How Do I Undo Changes?

Committed changes and tracked changes.

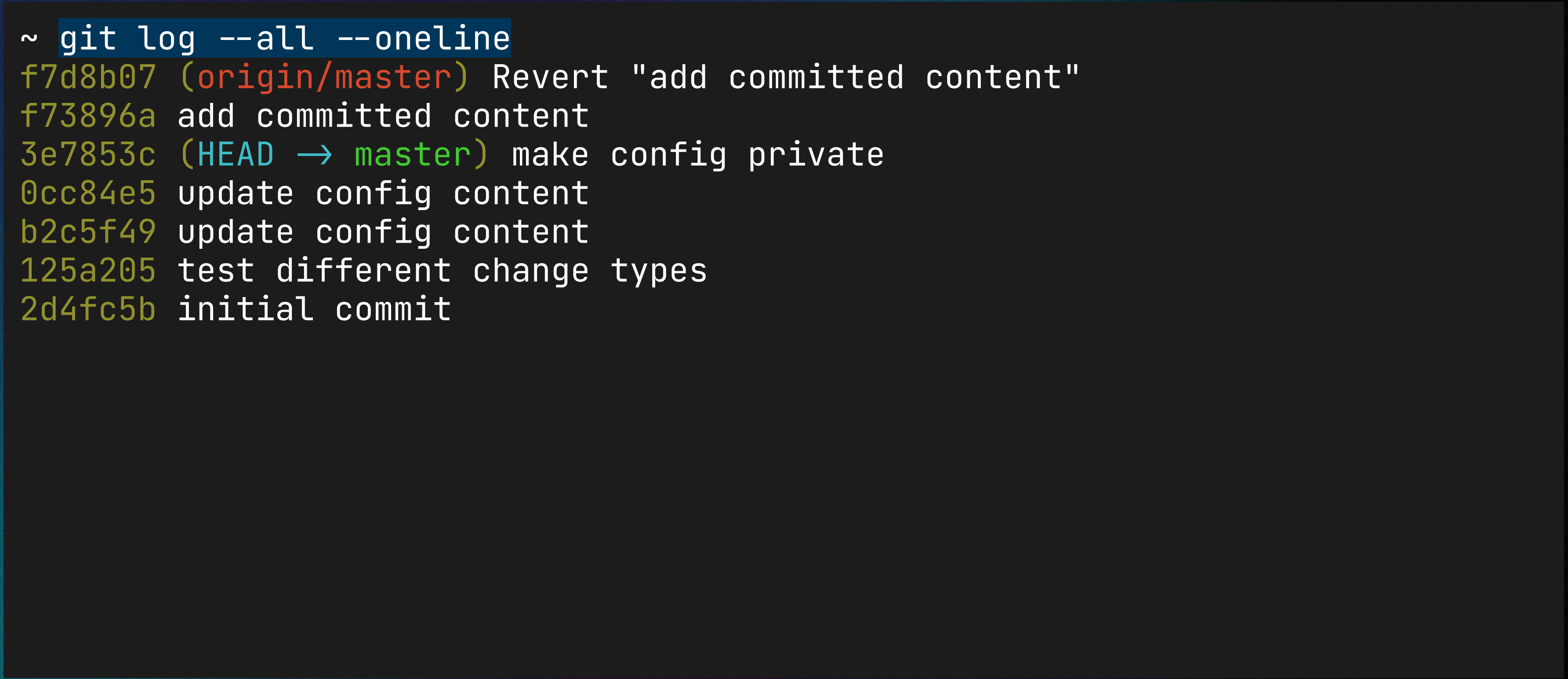
git reset --hard <commit hash to move to>



```
~ git reset --hard HEAD~
HEAD is now at 3e7853c make config private
~ git status
On branch master
Your branch is behind 'origin/master' by 2 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)

nothing to commit, working tree clean
~ git log -3 --oneline
3e7853c (HEAD → master) make config private
0cc84e5 update config content
b2c5f49 update config content
```

How Do I Display Commits?

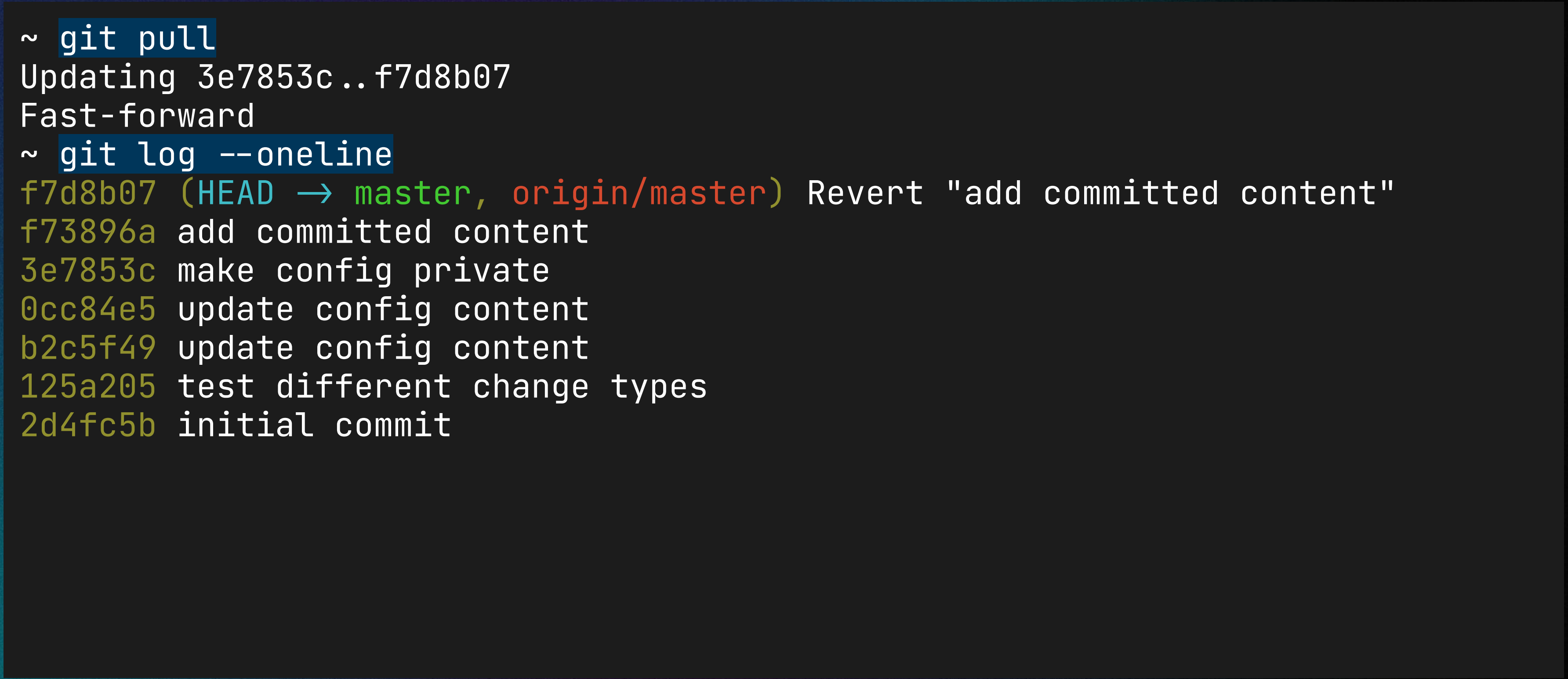
Not related to HEAD.



```
~ git log --all --oneline
f7d8b07 (origin/master) Revert "add committed content"
f73896a add committed content
3e7853c (HEAD -> master) make config private
0cc84e5 update config content
b2c5f49 update config content
125a205 test different change types
2d4fc5b initial commit
```

How Do I Download Changes From Remote?

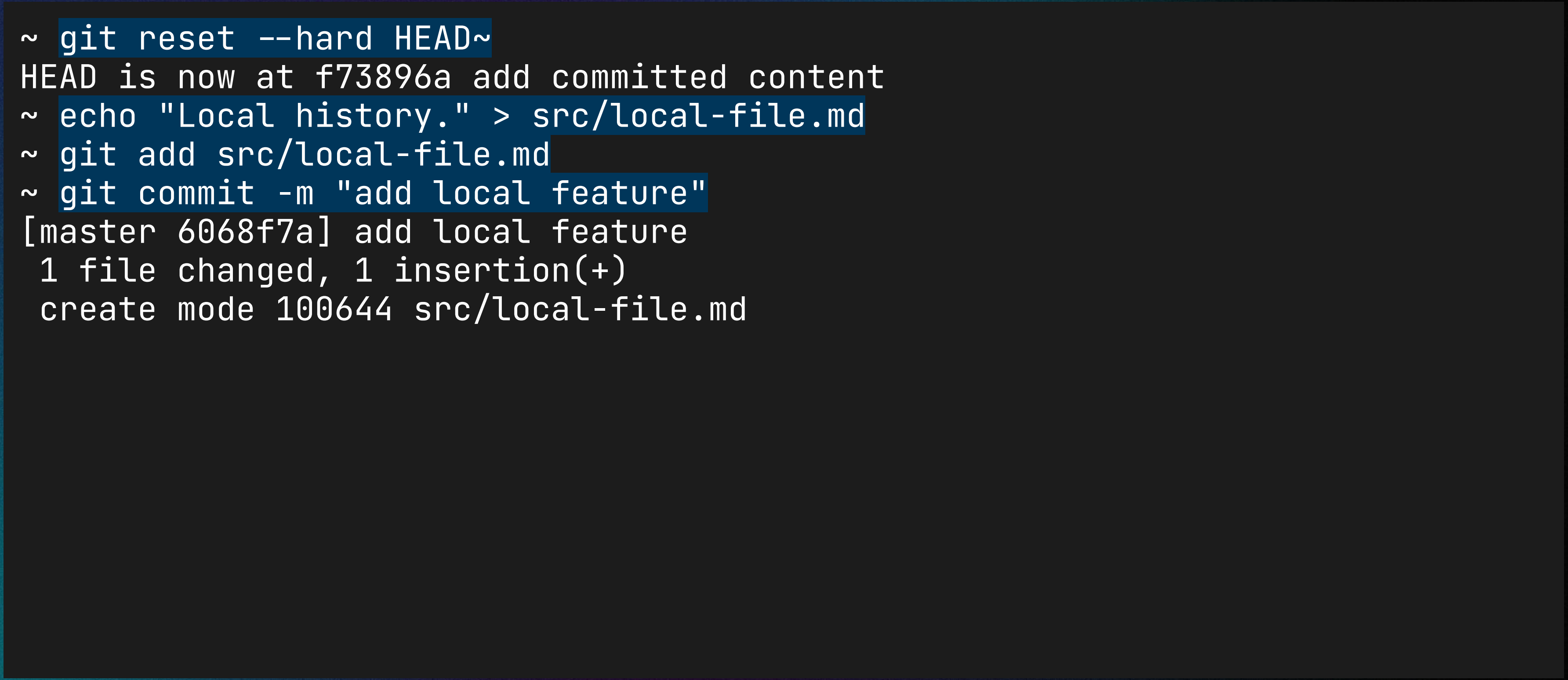
`git pull`



```
~ git pull
Updating 3e7853c..f7d8b07
Fast-forward
~ git log --oneline
f7d8b07 (HEAD -> master, origin/master) Revert "add committed content"
f73896a add committed content
3e7853c make config private
0cc84e5 update config content
b2c5f49 update config content
125a205 test different change types
2d4fc5b initial commit
```

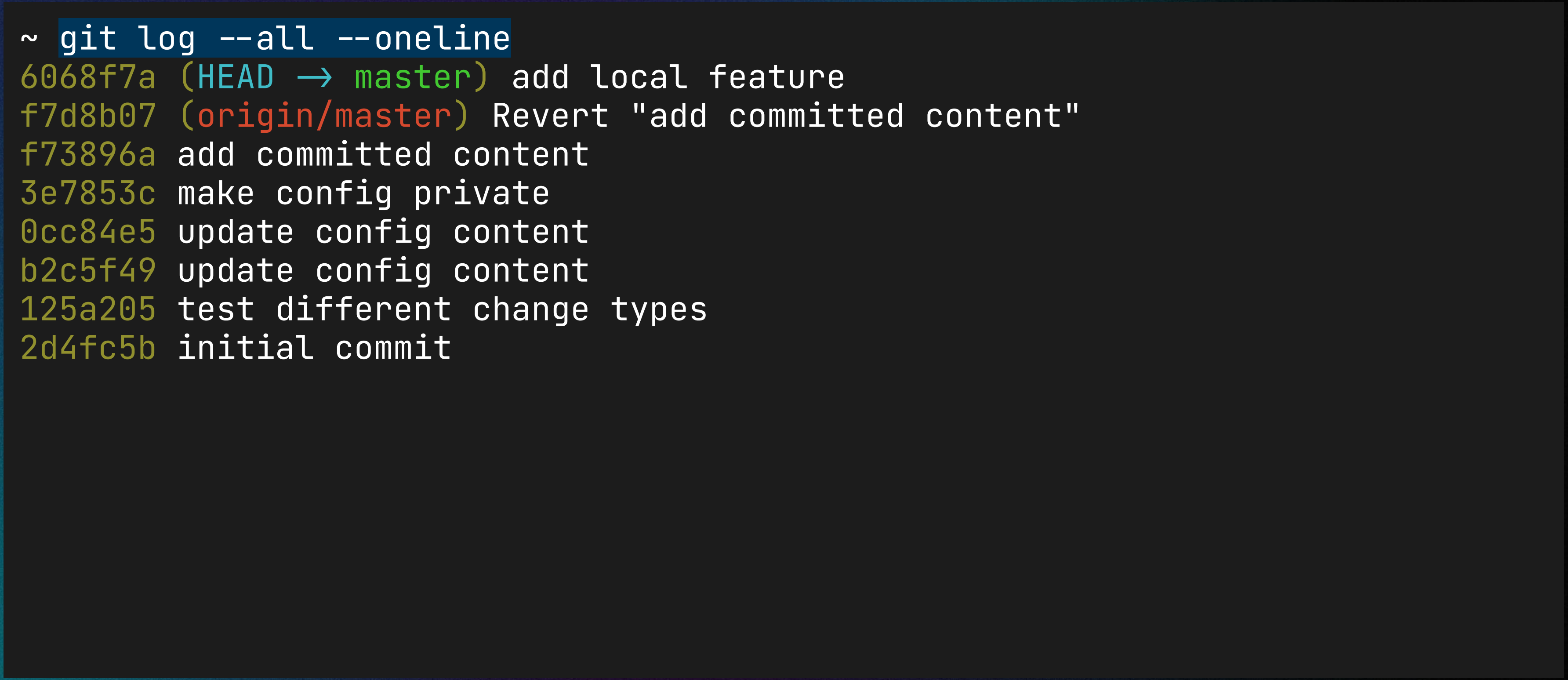
Prep Content

Diverge histories.



```
~ git reset --hard HEAD~
HEAD is now at f73896a add committed content
~ echo "Local history." > src/local-file.md
~ git add src/local-file.md
~ git commit -m "add local feature"
[master 6068f7a] add local feature
 1 file changed, 1 insertion(+)
 create mode 100644 src/local-file.md
```

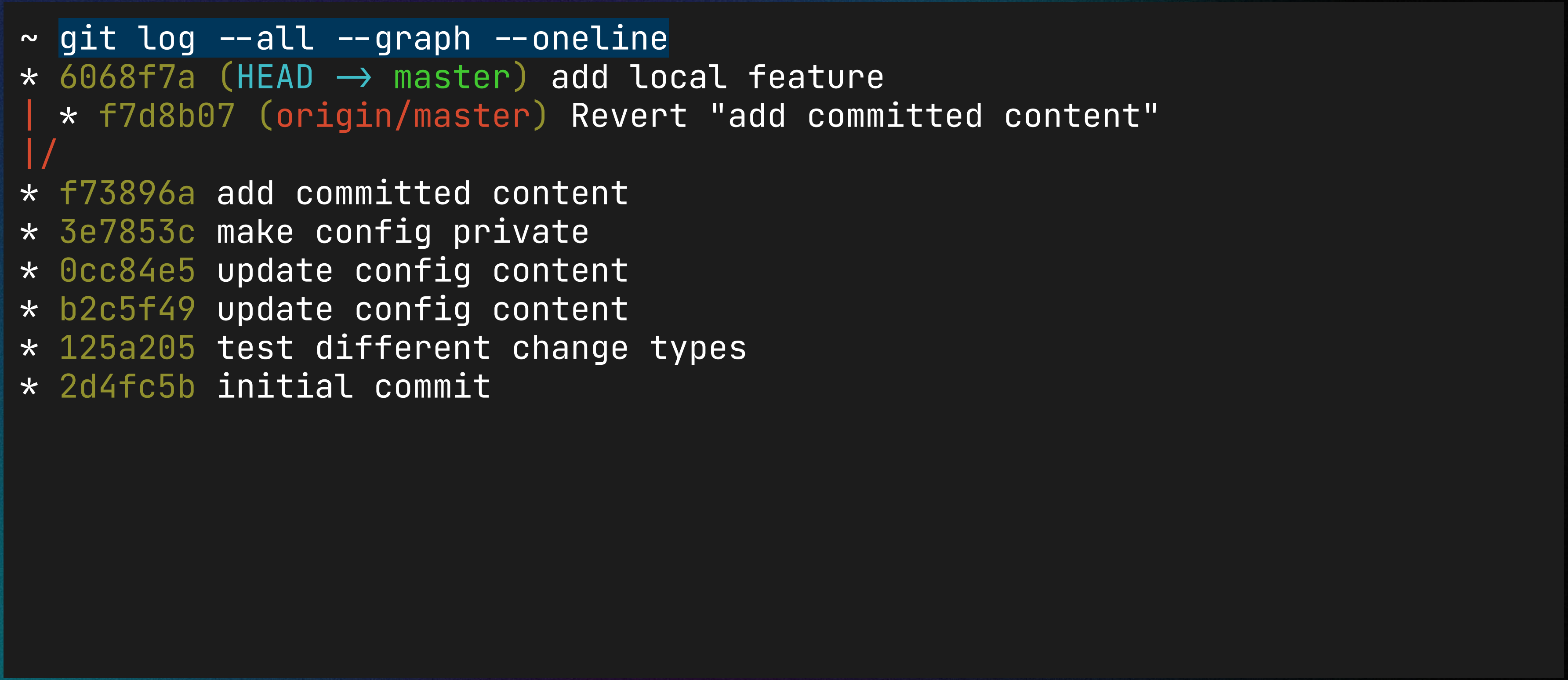
Warning: Displaying Missing Commits.



```
~ git log --all --oneline
6068f7a (HEAD → master) add local feature
f7d8b07 (origin/master) Revert "add committed content"
f73896a add committed content
3e7853c make config private
0cc84e5 update config content
b2c5f49 update config content
125a205 test different change types
2d4fc5b initial commit
```

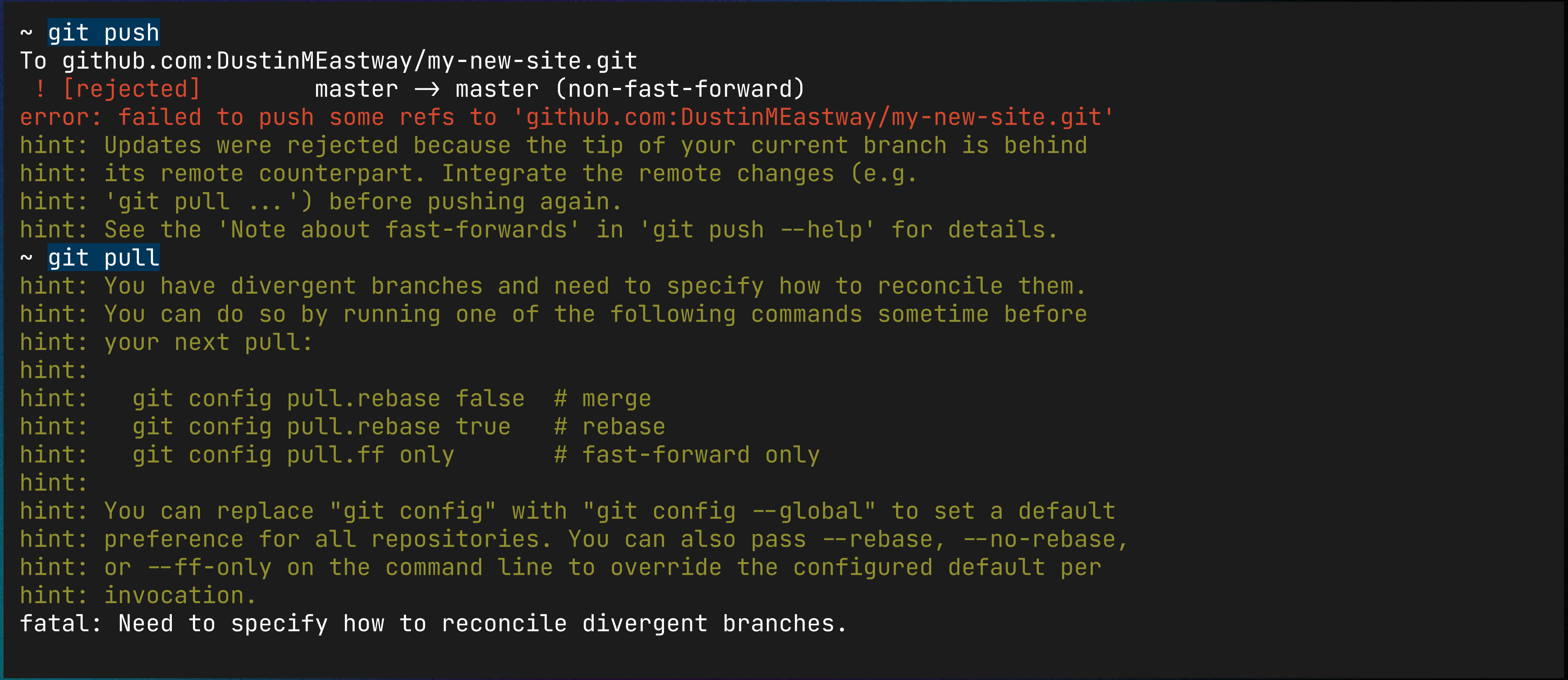
How Do I Display Commits?

With a graph.



```
~ git log --all --graph --oneline
* 6068f7a (HEAD -> master) add local feature
| * f7d8b07 (origin/master) Revert "add committed content"
|/
* f73896a add committed content
* 3e7853c make config private
* 0cc84e5 update config content
* b2c5f49 update config content
* 125a205 test different change types
* 2d4fc5b initial commit
```

Warning: Diverging Histories

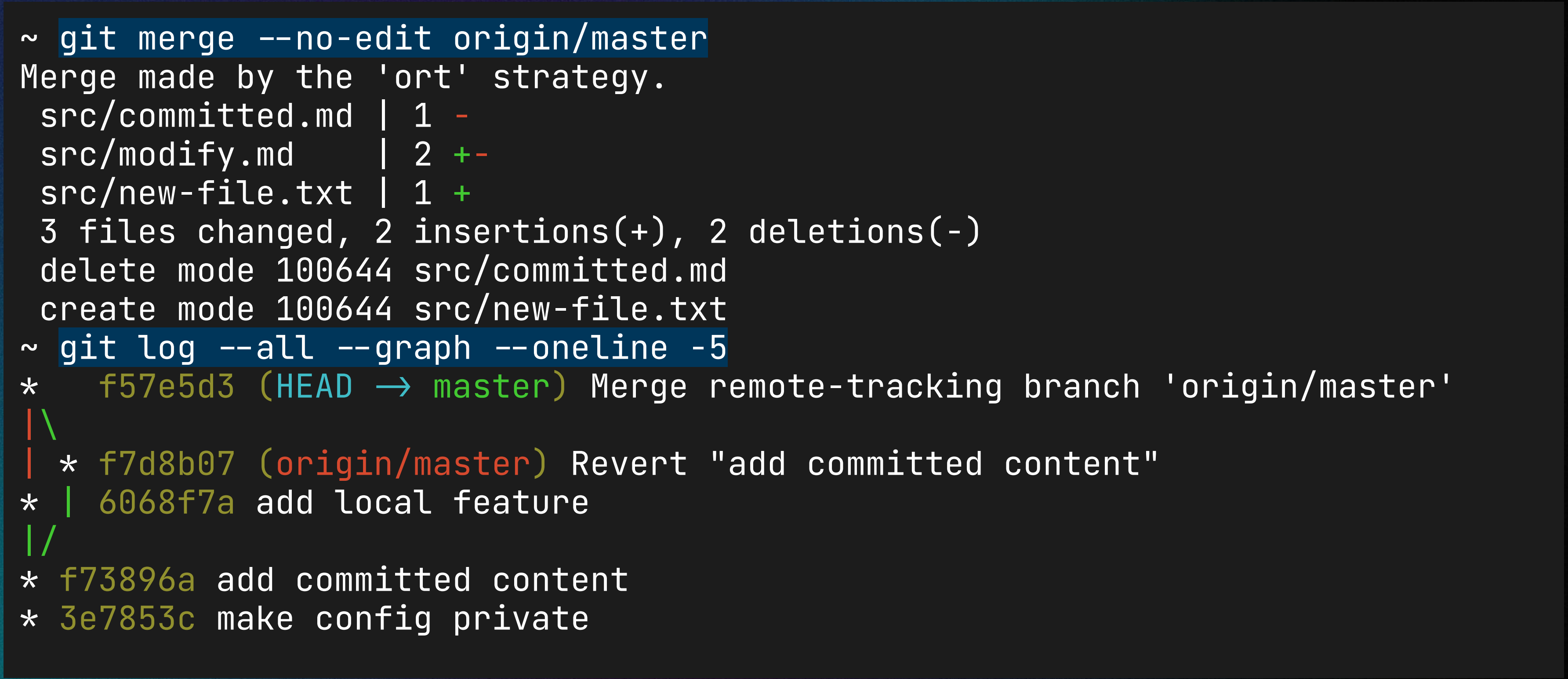


```
~ git push
To github.com:DustinMEastway/my-new-site.git
 ! [rejected]        master -> master (non-fast-forward)
error: failed to push some refs to 'github.com:DustinMEastway/my-new-site.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
~ git pull
hint: You have divergent branches and need to specify how to reconcile them.
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:
hint:   git config pull.rebase false  # merge
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only       # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
fatal: Need to specify how to reconcile divergent branches.
```

How Do I Combine Histories?

Merge.

git merge [--no-edit] <full branch name>

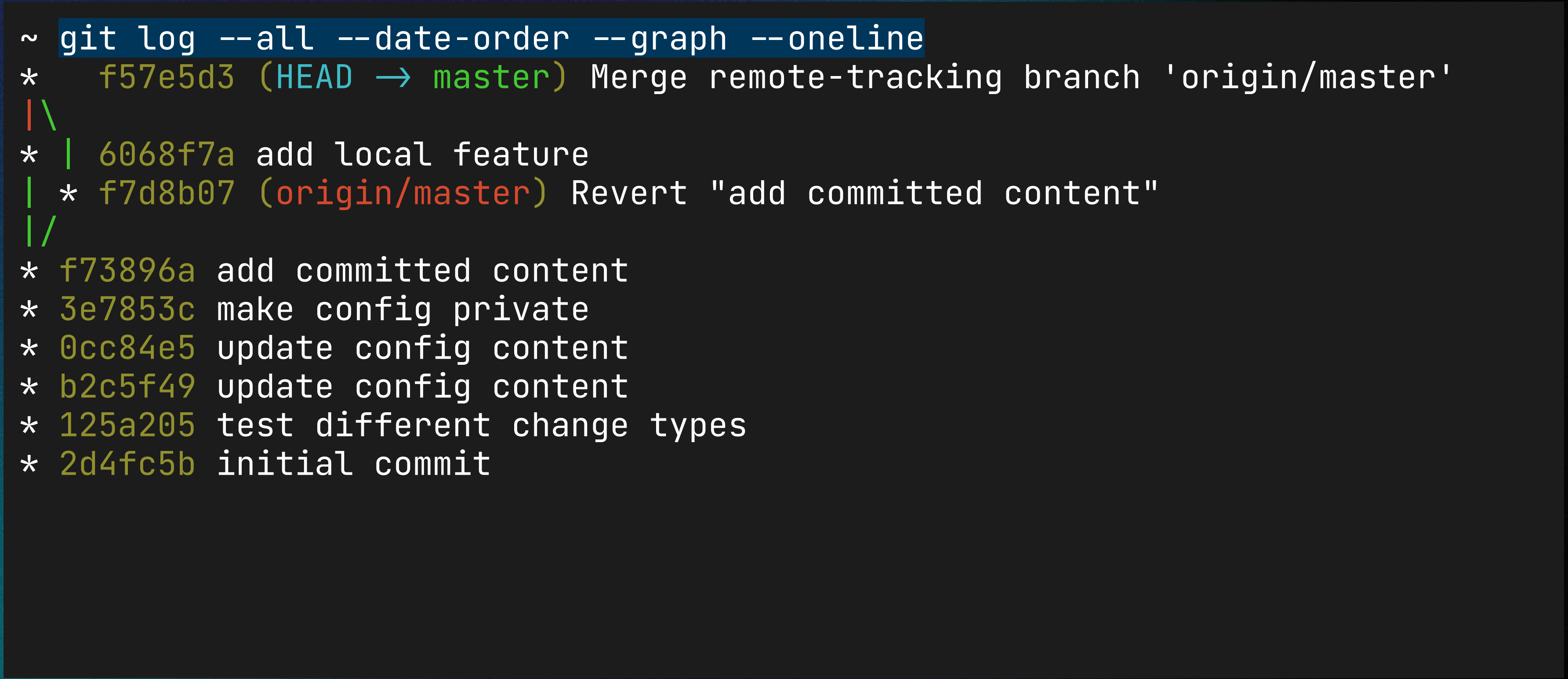


```
~ git merge --no-edit origin/master
Merge made by the 'ort' strategy.
 src/committed.md | 1 -
 src/modify.md    | 2 +-
 src/new-file.txt | 1 +
 3 files changed, 2 insertions(+), 2 deletions(-)
 delete mode 100644 src/committed.md
 create mode 100644 src/new-file.txt
~ git log --all --graph --oneline -5
*   f57e5d3 (HEAD → master) Merge remote-tracking branch 'origin/master'
|\
| * f7d8b07 (origin/master) Revert "add committed content"
* | 6068f7a add local feature
|/
* f73896a add committed content
* 3e7853c make config private
```

How Do I Display Commits?

With a graph ordered by date.

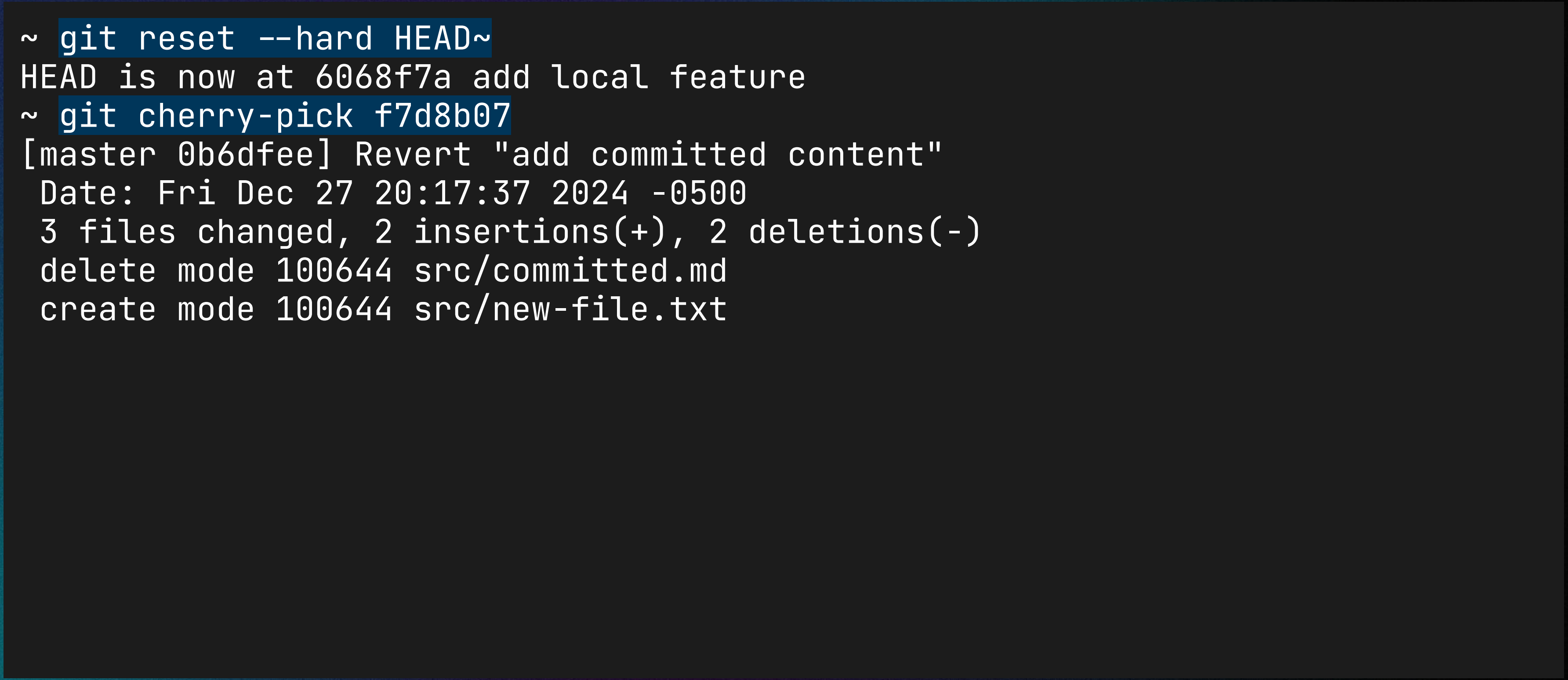
`git log --all --date-order --graph --oneline`



```
~ git log --all --date-order --graph --oneline
*   f57e5d3 (HEAD → master) Merge remote-tracking branch 'origin/master'
|\
* | 6068f7a add local feature
| * f7d8b07 (origin/master) Revert "add committed content"
|/
* f73896a add committed content
* 3e7853c make config private
* 0cc84e5 update config content
* b2c5f49 update config content
* 125a205 test different change types
* 2d4fc5b initial commit
```

How Do I Take A Commit From Elsewhere?

git cherry-pick <commit hash>

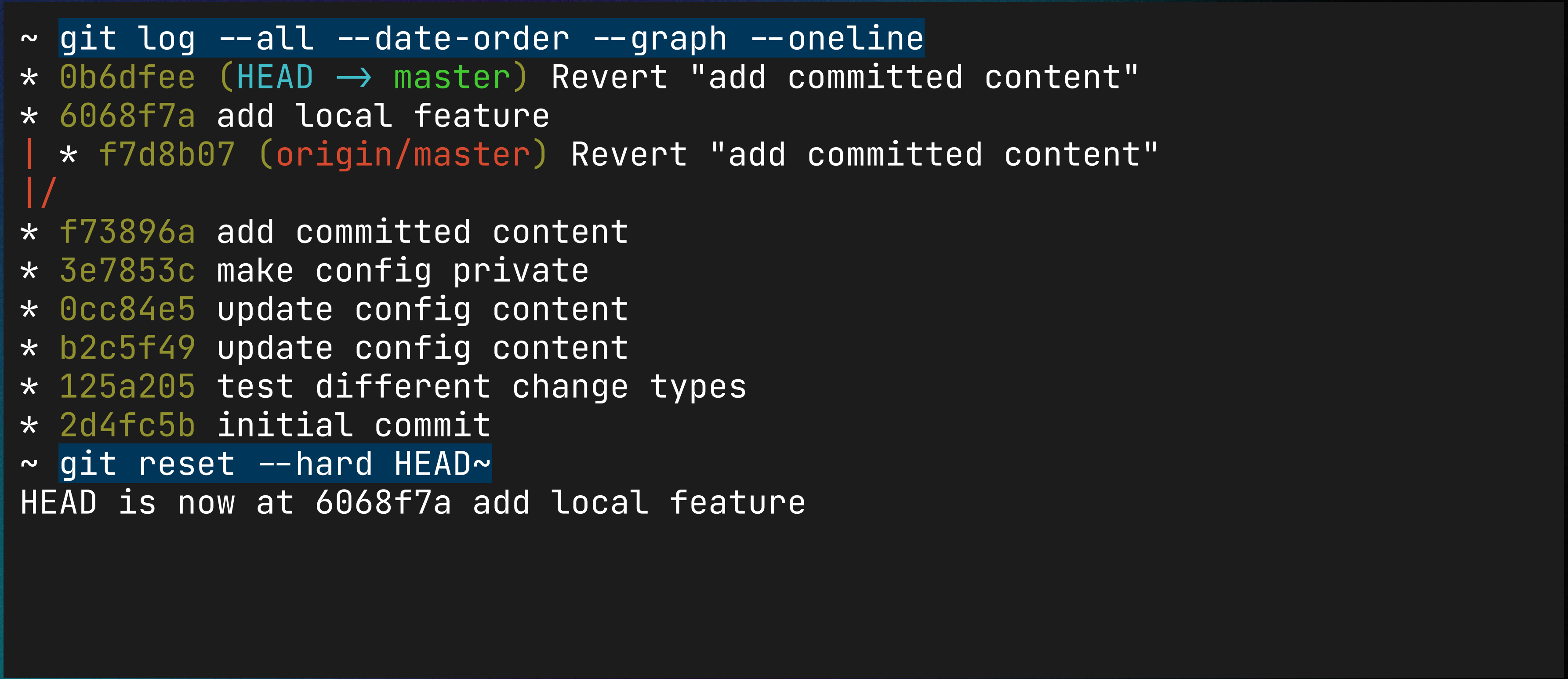


```
~ git reset --hard HEAD~
HEAD is now at 6068f7a add local feature
~ git cherry-pick f7d8b07
[master 0b6dfee] Revert "add committed content"
 Date: Fri Dec 27 20:17:37 2024 -0500
 3 files changed, 2 insertions(+), 2 deletions(-)
 delete mode 100644 src/committed.md
 create mode 100644 src/new-file.txt
```

How Do I Take A Commit From Elsewhere?

Continued…

git cherry-pick <commit hash>

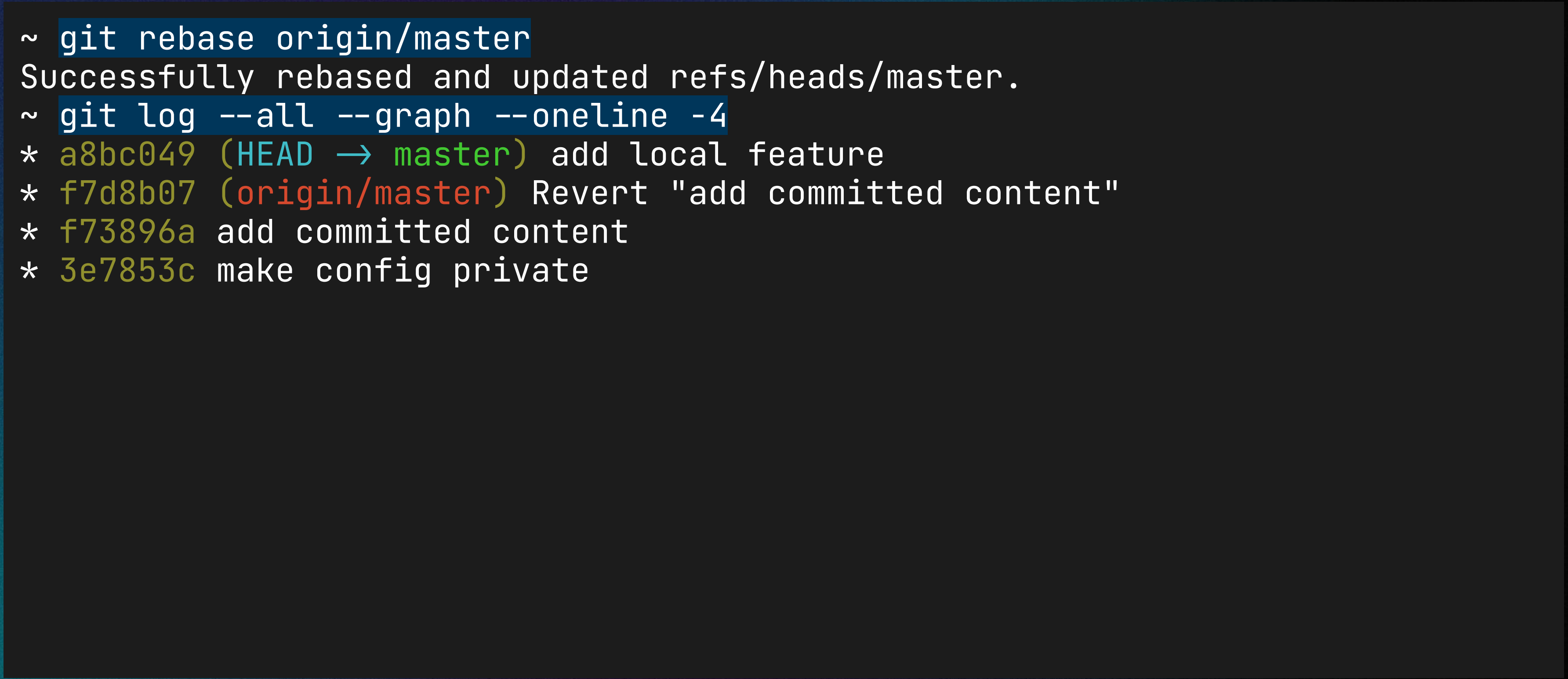


```
~ git log --all --date-order --graph --oneline
* 0b6dfee (HEAD → master) Revert "add committed content"
* 6068f7a add local feature
| * f7d8b07 (origin/master) Revert "add committed content"
|/
* f73896a add committed content
* 3e7853c make config private
* 0cc84e5 update config content
* b2c5f49 update config content
* 125a205 test different change types
* 2d4fc5b initial commit
~ git reset --hard HEAD~
HEAD is now at 6068f7a add local feature
```

How Do I Combine Histories?

Rebase.

git rebase <full branch name or commit hash>



```
~ git rebase origin/master
Successfully rebased and updated refs/heads/master.
~ git log --all --graph --oneline -4
* a8bc049 (HEAD -> master) add local feature
* f7d8b07 (origin/master) Revert "add committed content"
* f73896a add committed content
* 3e7853c make config private
```

Note: Comparing Histories.

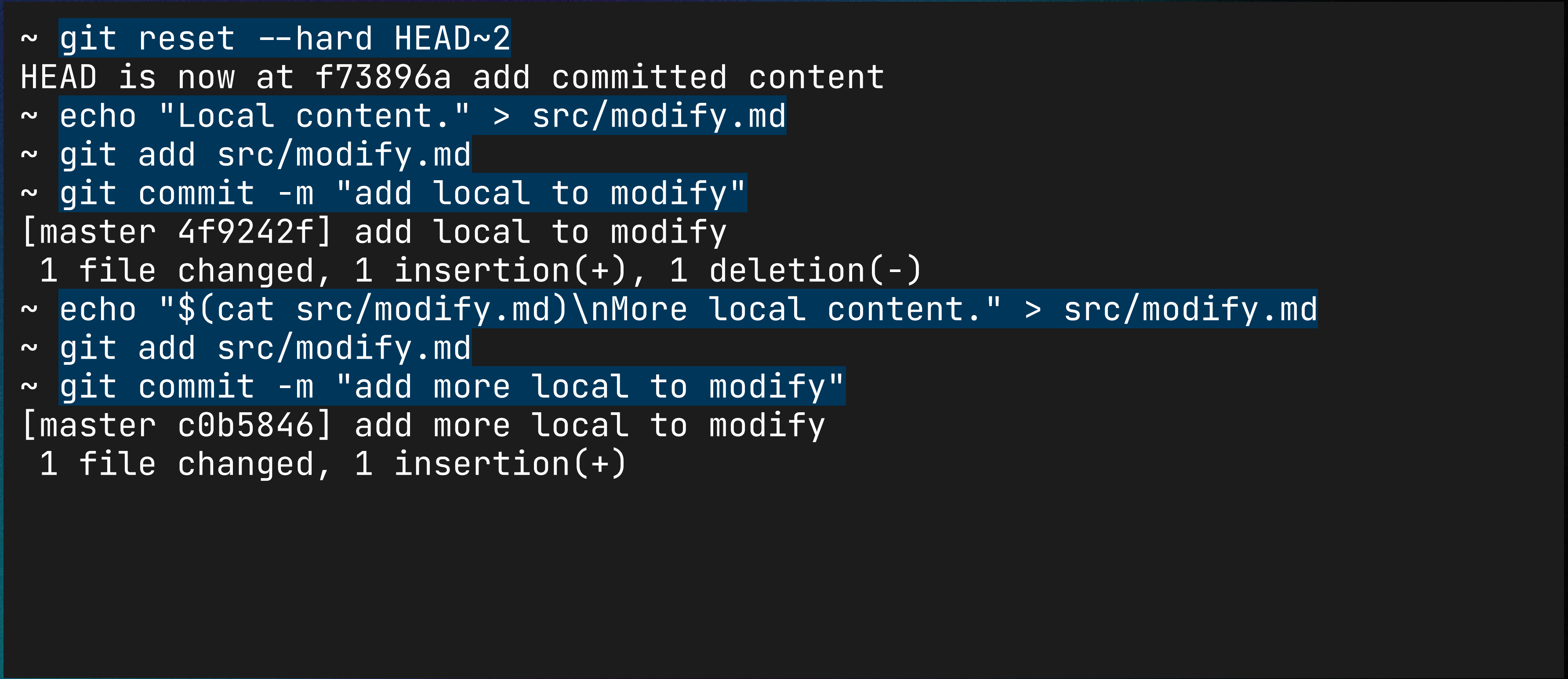
```
Original
* 6068f7a (HEAD -> master) add local feature
| * f7d8b07 (origin/master) Revert "add committed content"
|/
* f73896a add committed content
* 3e7853c make config private
```

```
Merging
*   f57e5d3 (HEAD -> master) Merge remote-tracking branch 'origin/master'
|\
* | 6068f7a add local feature
| * f7d8b07 (origin/master) Revert "add committed content"
|/
* f73896a add committed content
* 3e7853c make config private
```

```
Rebasing
* a8bc049 (HEAD -> master) add local feature
* f7d8b07 (origin/master) Revert "add committed content"
* f73896a add committed content
* 3e7853c make config private
```

Prep Content

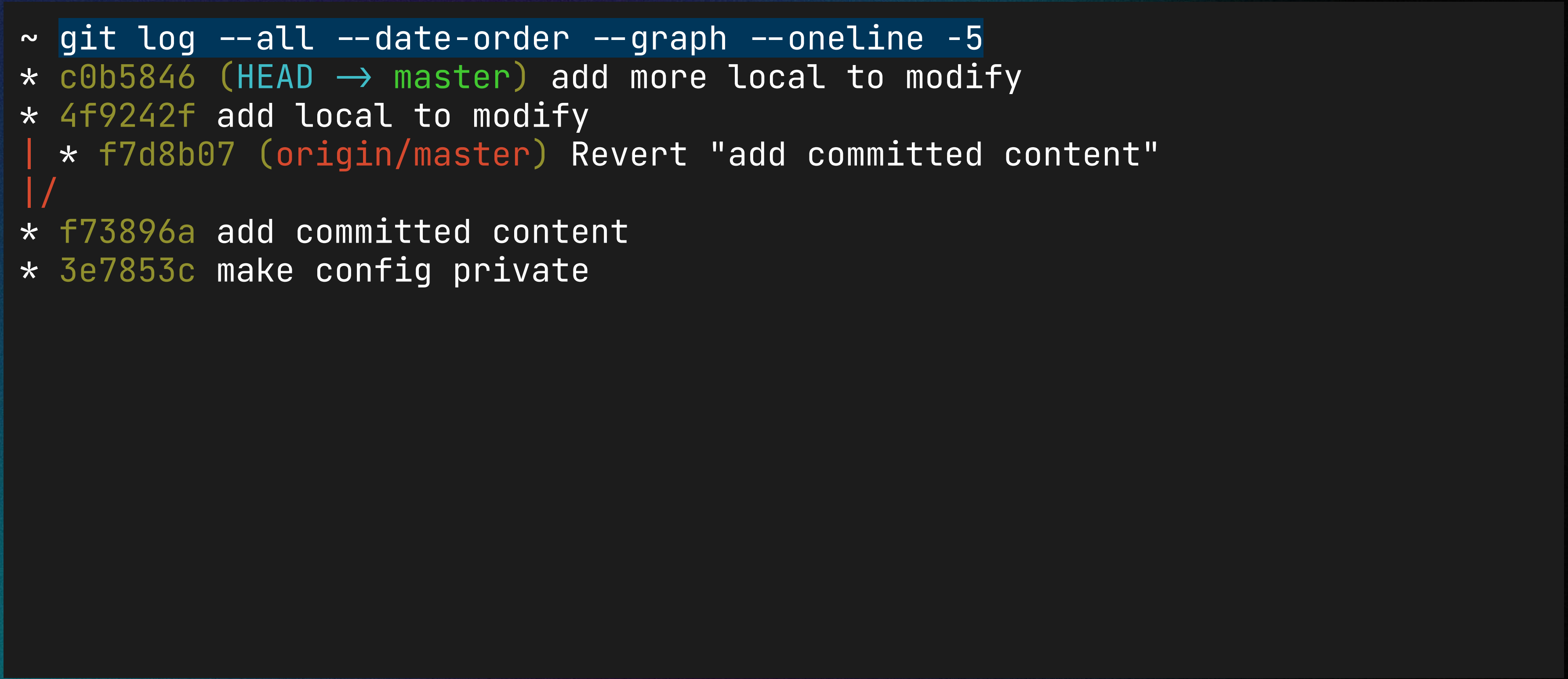
Diverge histories with conflicts.



```
~ git reset --hard HEAD~2
HEAD is now at f73896a add committed content
~ echo "Local content." > src/modify.md
~ git add src/modify.md
~ git commit -m "add local to modify"
[master 4f9242f] add local to modify
 1 file changed, 1 insertion(+), 1 deletion(-)
~ echo "$(cat src/modify.md)\nMore local content." > src/modify.md
~ git add src/modify.md
~ git commit -m "add more local to modify"
[master c0b5846] add more local to modify
 1 file changed, 1 insertion(+)
```

Prep Content

Diverge histories with conflicts. Continued...

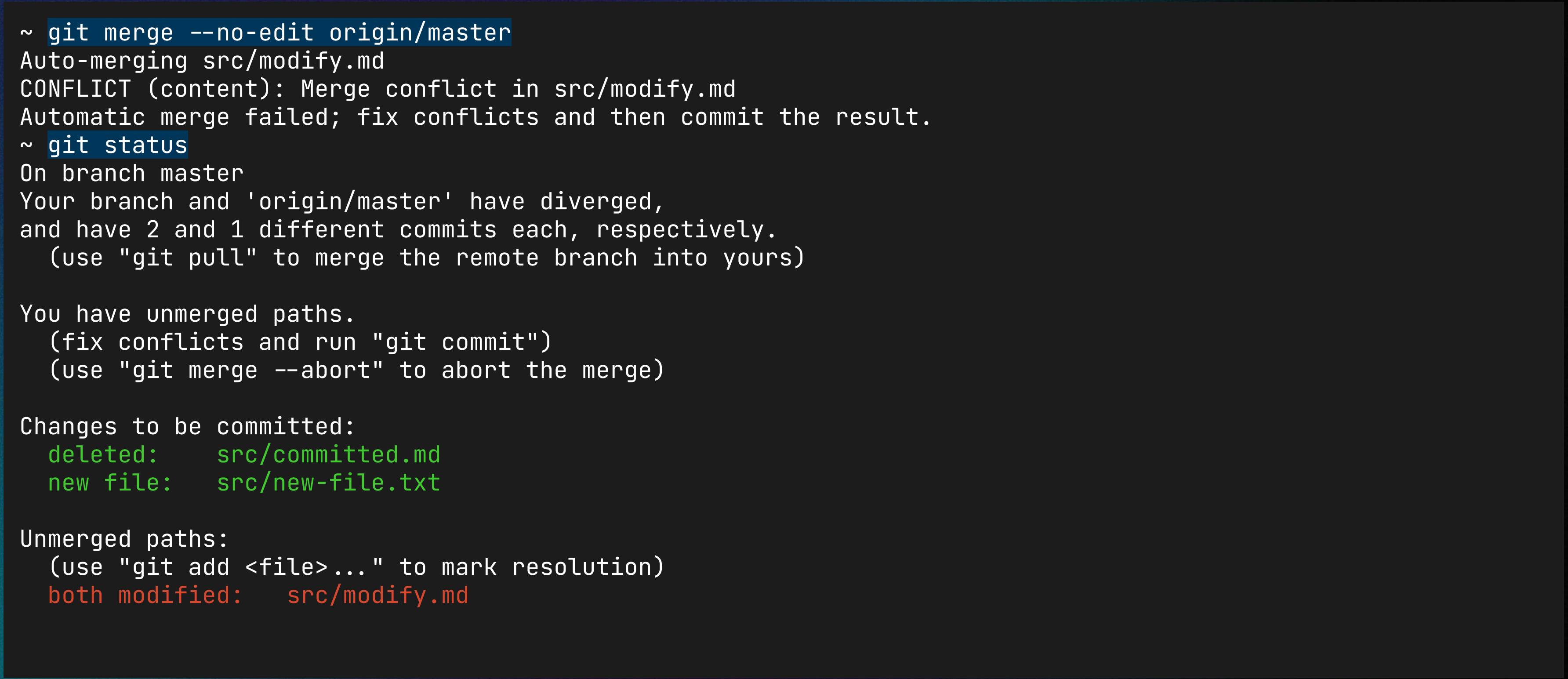


```
~ git log --all --date-order --graph --oneline -5
* c0b5846 (HEAD -> master) add more local to modify
* 4f9242f add local to modify
| * f7d8b07 (origin/master) Revert "add committed content"
|/
* f73896a add committed content
* 3e7853c make config private
```

How Do I Combine Histories?

Merge conflicts.

git merge [--no-edit] <full branch name>



```
~ git merge --no-edit origin/master
Auto-merging src/modify.md
CONFLICT (content): Merge conflict in src/modify.md
Automatic merge failed; fix conflicts and then commit the result.
~ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 2 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  deleted:    src/committed.md
  new file:   src/new-file.txt

Unmerged paths:
  (use "git add <file>..." to mark resolution)
  both modified:   src/modify.md
```

How Do I Combine Histories?

Merge conflicts. Continued...

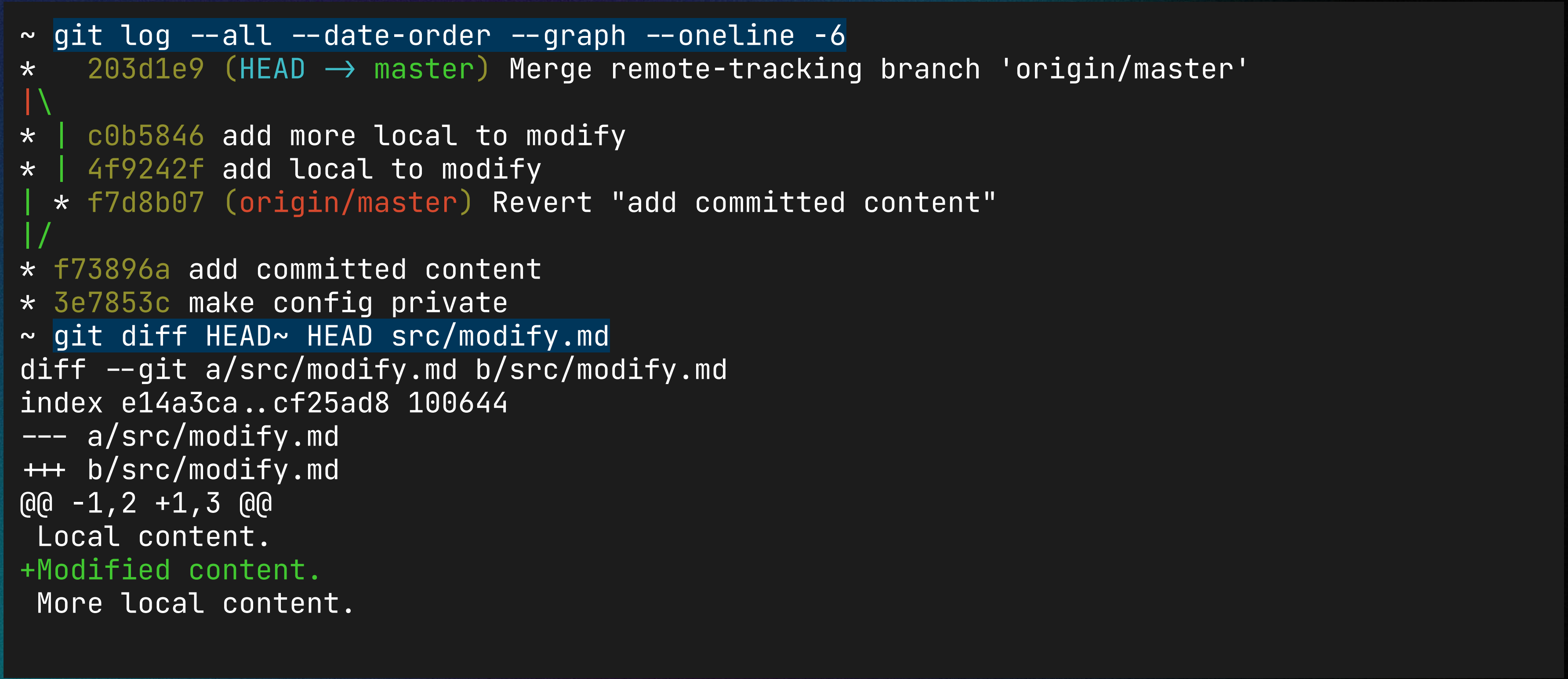
git merge [--no-edit] <full branch name>

```
~ cat src/modify.md
<<<<<<< HEAD
Local content.
More local content.
=======
Modified content.
>>>>>>> origin/master
~ echo "Local content.\nModified content.\nMore local content." > src/modify.md
~ git add src/modify.md
~ git commit --no-edit
[master 203d1e9] Merge remote-tracking branch 'origin/master'
```

How Do I Combine Histories?

Merge conflicts. Continued...

git merge [--no-edit] <full branch name>

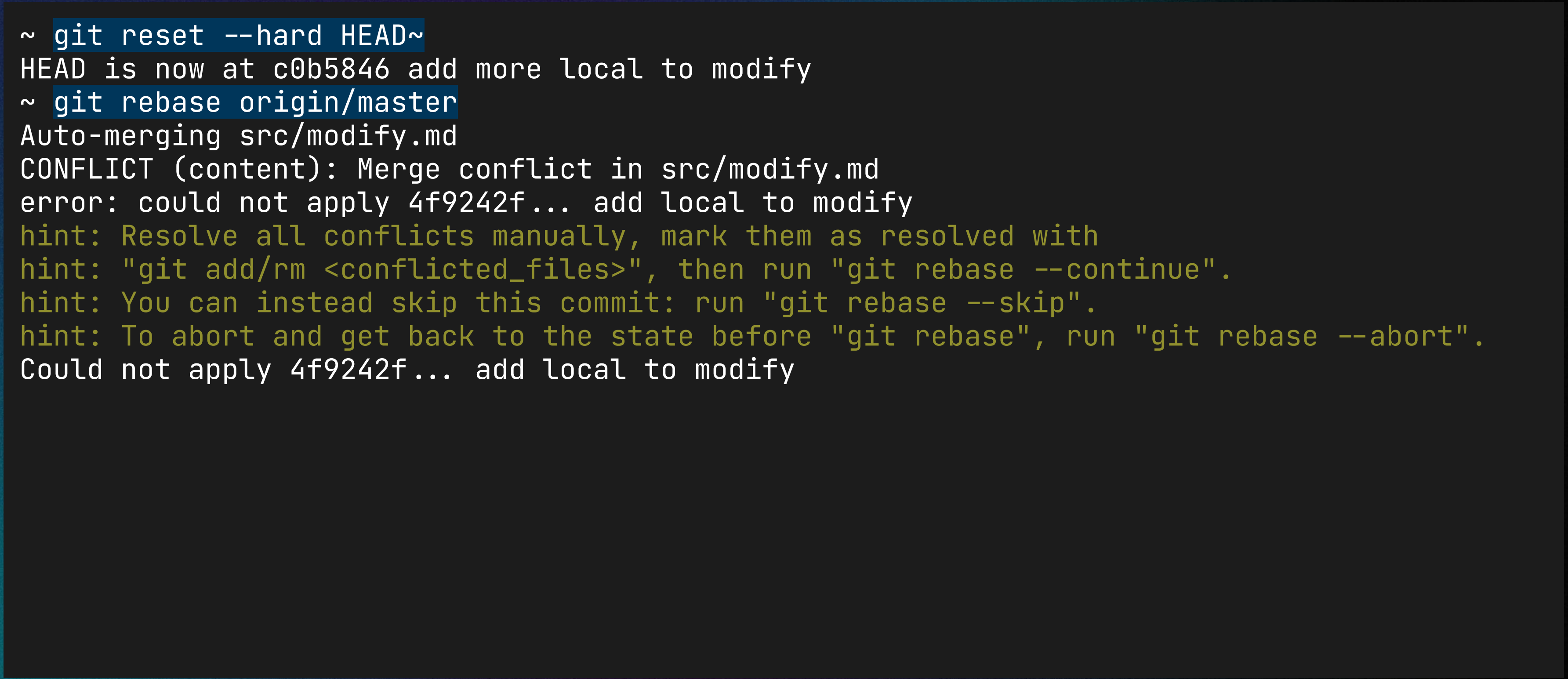


```
~ git log --all --date-order --graph --oneline -6
*   203d1e9 (HEAD → master) Merge remote-tracking branch 'origin/master'
|\
* | c0b5846 add more local to modify
* | 4f9242f add local to modify
| * f7d8b07 (origin/master) Revert "add committed content"
|/
* f73896a add committed content
* 3e7853c make config private
~ git diff HEAD~ HEAD src/modify.md
diff --git a/src/modify.md b/src/modify.md
index e14a3ca..cf25ad8 100644
--- a/src/modify.md
+++ b/src/modify.md
@@ -1,2 +1,3 @@
 Local content.
+Modified content.
 More local content.
```

How Do I Combine Histories?

Rebase conflicts.

git rebase <full branch name or commit hash>

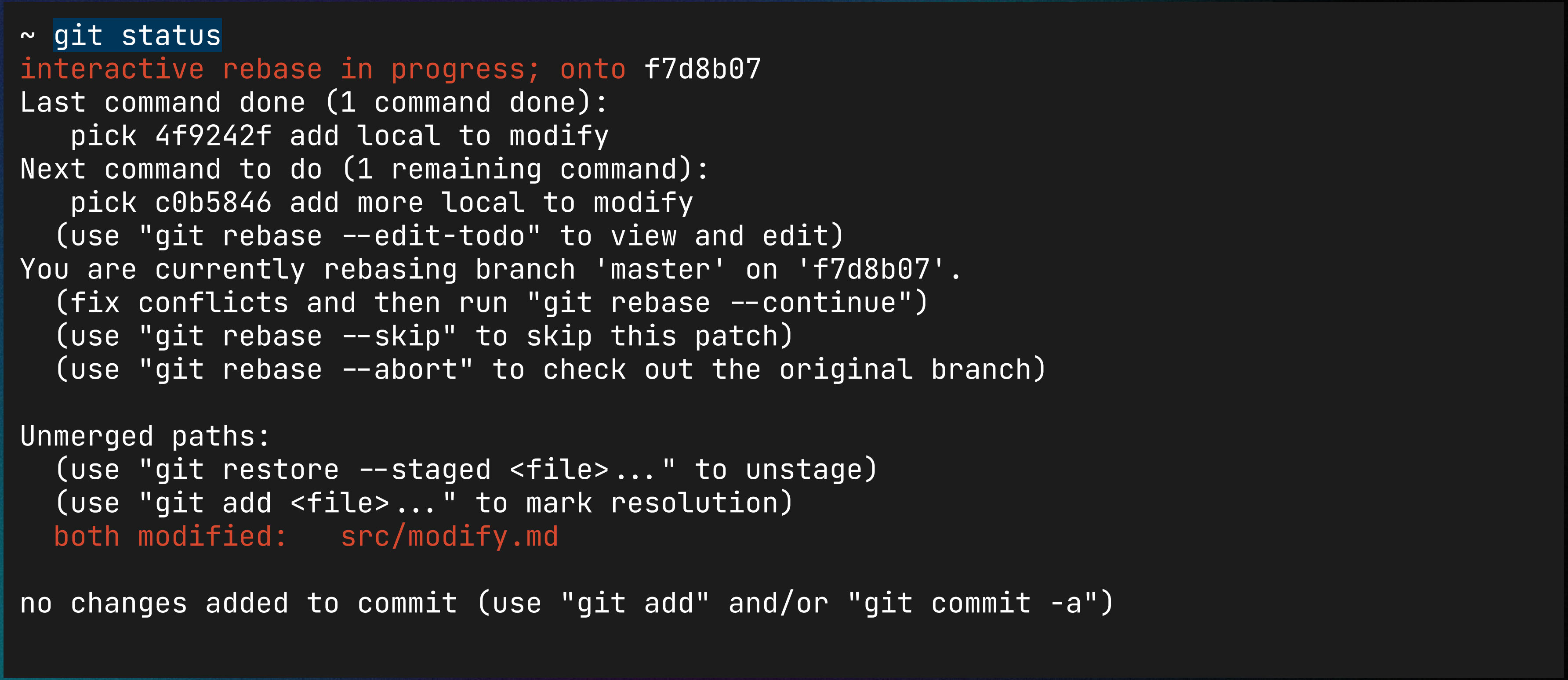


```
~ git reset --hard HEAD~
HEAD is now at c0b5846 add more local to modify
~ git rebase origin/master
Auto-merging src/modify.md
CONFLICT (content): Merge conflict in src/modify.md
error: could not apply 4f9242f... add local to modify
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply 4f9242f... add local to modify
```

How Do I Combine Histories?

Rebase conflicts. Continued...

`git rebase <full branch name or commit hash>`

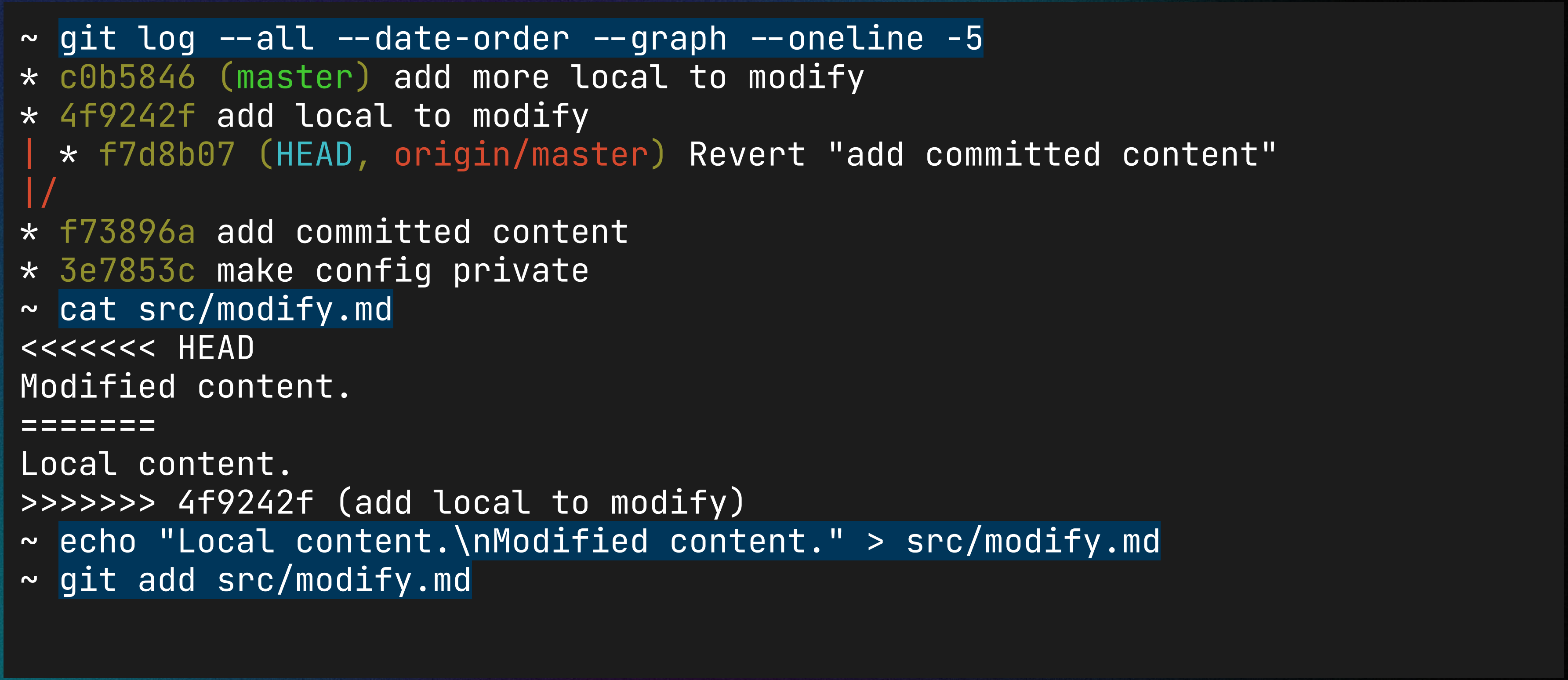


```
~ git status
interactive rebase in progress; onto f7d8b07
Last command done (1 command done):
   pick 4f9242f add local to modify
Next command to do (1 remaining command):
   pick c0b5846 add more local to modify
  (use "git rebase --edit-todo" to view and edit)
You are currently rebasing branch 'master' on 'f7d8b07'.
  (fix conflicts and then run "git rebase --continue")
  (use "git rebase --skip" to skip this patch)
  (use "git rebase --abort" to check out the original branch)

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
	both modified:   src/modify.md

no changes added to commit (use "git add" and/or "git commit -a")
```

How Do I Combine Histories?

Rebase conflicts. Continued...

git rebase <full branch name or commit hash>

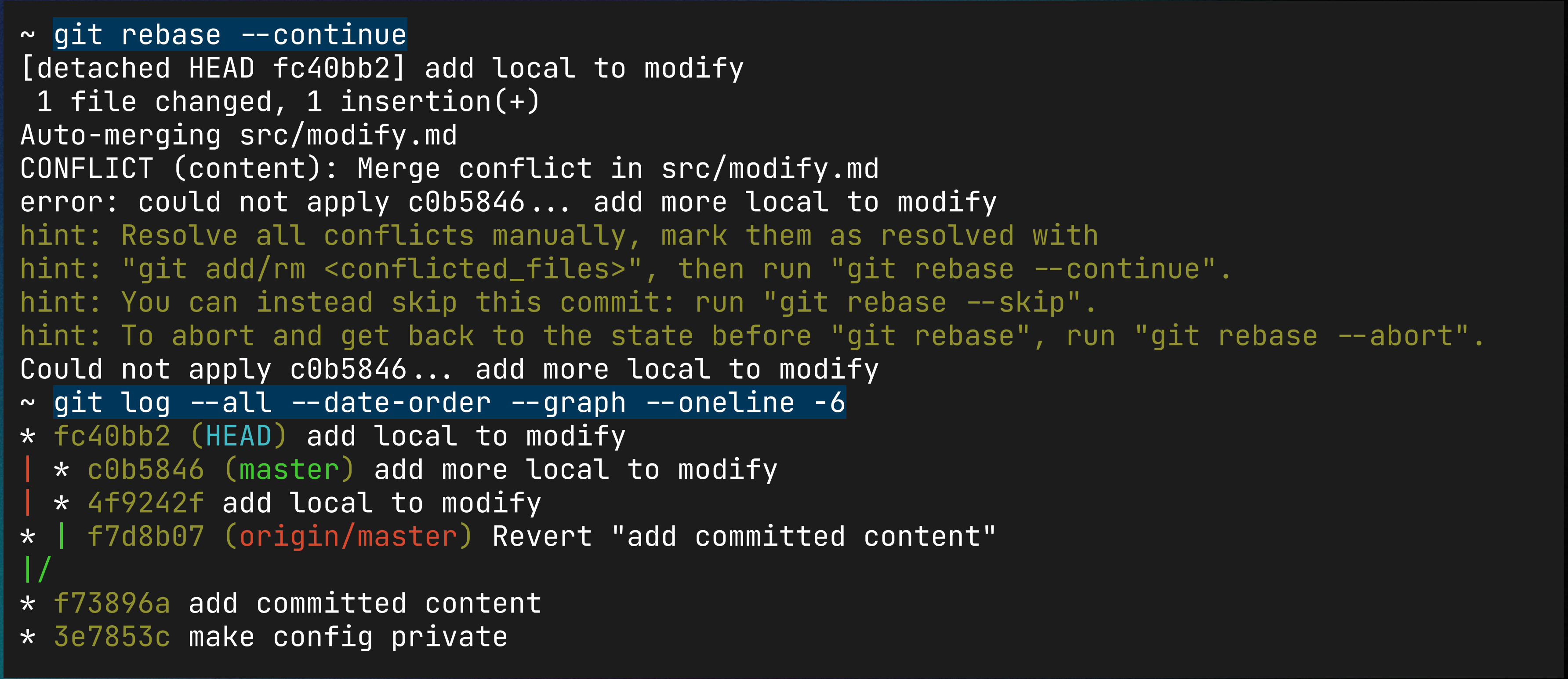


```
~ git log --all --date-order --graph --oneline -5
* c0b5846 (master) add more local to modify
* 4f9242f add local to modify
| * f7d8b07 (HEAD, origin/master) Revert "add committed content"
|/
* f73896a add committed content
* 3e7853c make config private
~ cat src/modify.md
<<<<<<< HEAD
Modified content.
=======
Local content.
>>>>>>> 4f9242f (add local to modify)
~ echo "Local content.\nModified content." > src/modify.md
~ git add src/modify.md
```

How Do I Combine Histories?

Rebase conflicts. Continued...

git rebase <full branch name or commit hash>

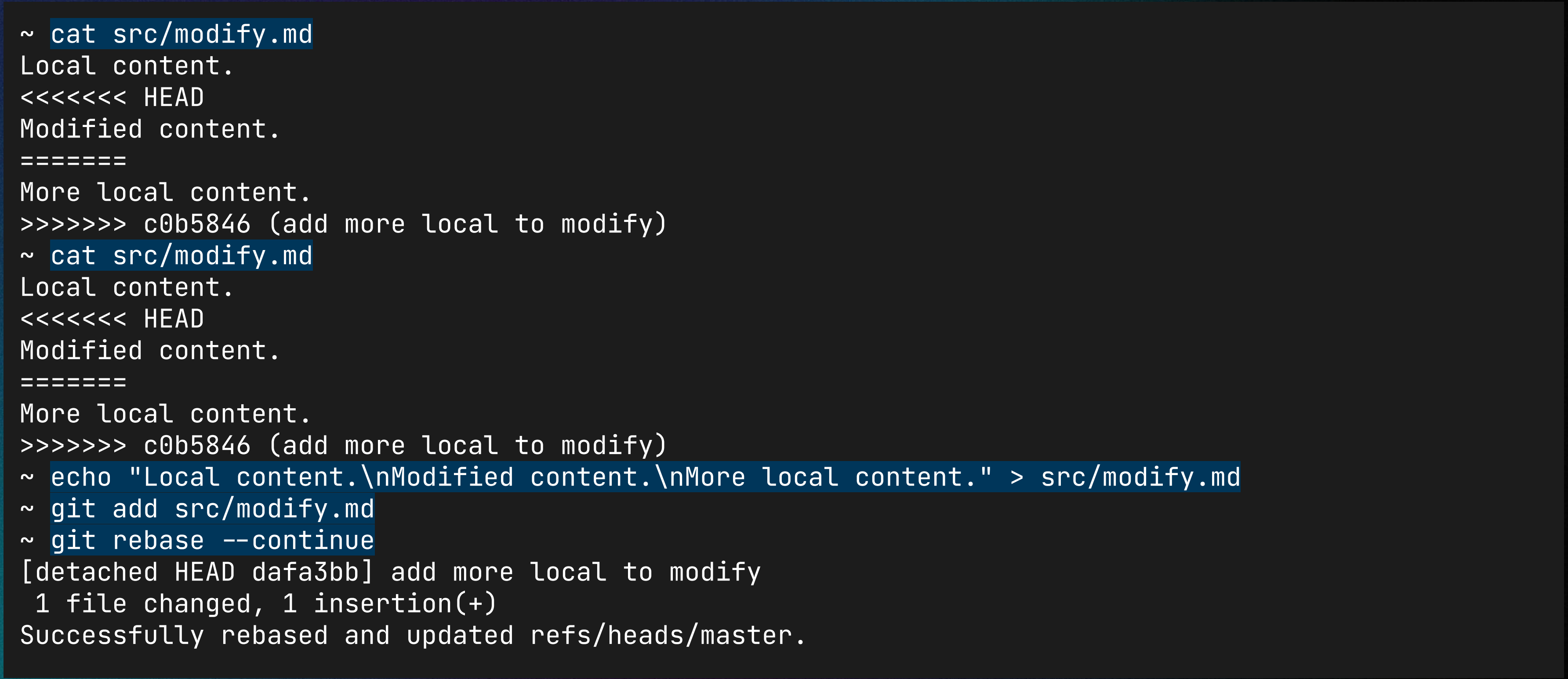


```
~ git rebase --continue
[detached HEAD fc40bb2] add local to modify
 1 file changed, 1 insertion(+)
Auto-merging src/modify.md
CONFLICT (content): Merge conflict in src/modify.md
error: could not apply c0b5846... add more local to modify
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply c0b5846... add more local to modify
~ git log --all --date-order --graph --oneline -6
* fc40bb2 (HEAD) add local to modify
| * c0b5846 (master) add more local to modify
| * 4f9242f add local to modify
* | f7d8b07 (origin/master) Revert "add committed content"
|/
* f73896a add committed content
* 3e7853c make config private
```

How Do I Combine Histories?

Rebase conflicts. Continued...

git rebase <full branch name or commit hash>

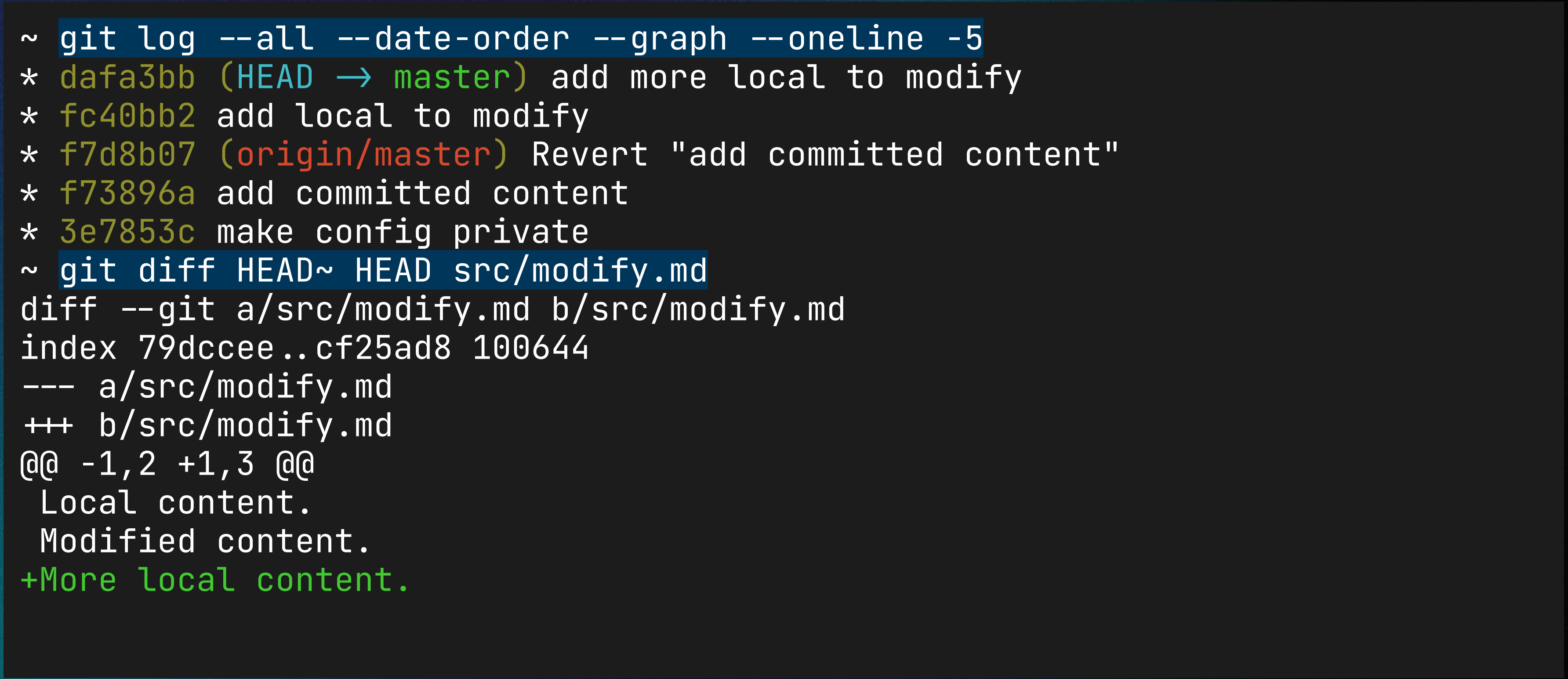


```
~ cat src/modify.md
Local content.
<<<<<<< HEAD
Modified content.
=======
More local content.
>>>>>>> c0b5846 (add more local to modify)
~ cat src/modify.md
Local content.
<<<<<<< HEAD
Modified content.
=======
More local content.
>>>>>>> c0b5846 (add more local to modify)
~ echo "Local content.\nModified content.\nMore local content." > src/modify.md
~ git add src/modify.md
~ git rebase --continue
[detached HEAD dafa3bb] add more local to modify
 1 file changed, 1 insertion(+)
Successfully rebased and updated refs/heads/master.
```

How Do I Combine Histories?

Rebase conflicts. Continued...

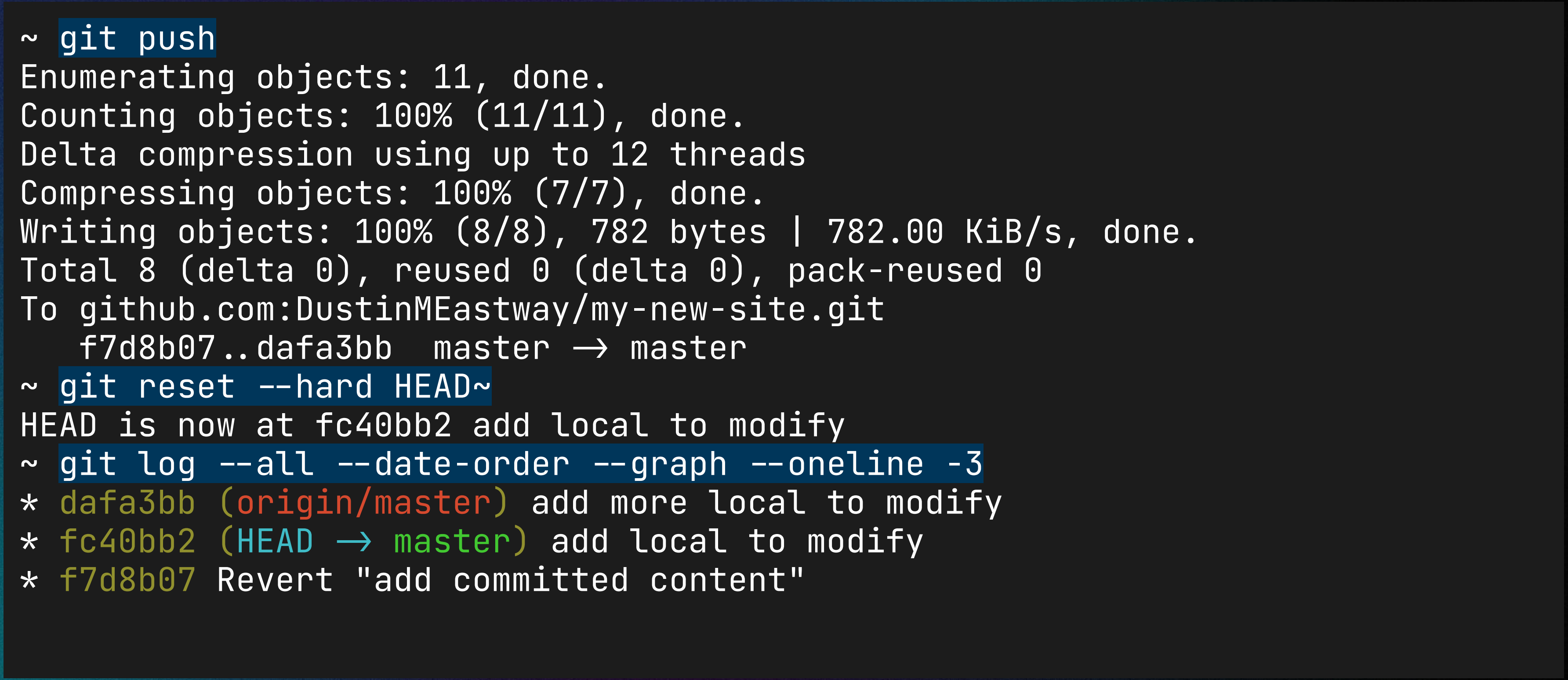
`git rebase <full branch name or commit hash>`



```
~ git log --all --date-order --graph --oneline -5
* dafa3bb (HEAD → master) add more local to modify
* fc40bb2 add local to modify
* f7d8b07 (origin/master) Revert "add committed content"
* f73896a add committed content
* 3e7853c make config private
~ git diff HEAD~ HEAD src/modify.md
diff --git a/src/modify.md b/src/modify.md
index 79dccee..cf25ad8 100644
--- a/src/modify.md
+++ b/src/modify.md
@@ -1,2 +1,3 @@
 Local content.
 Modified content.
+More local content.
```

Prep Content

Diverge histories again.

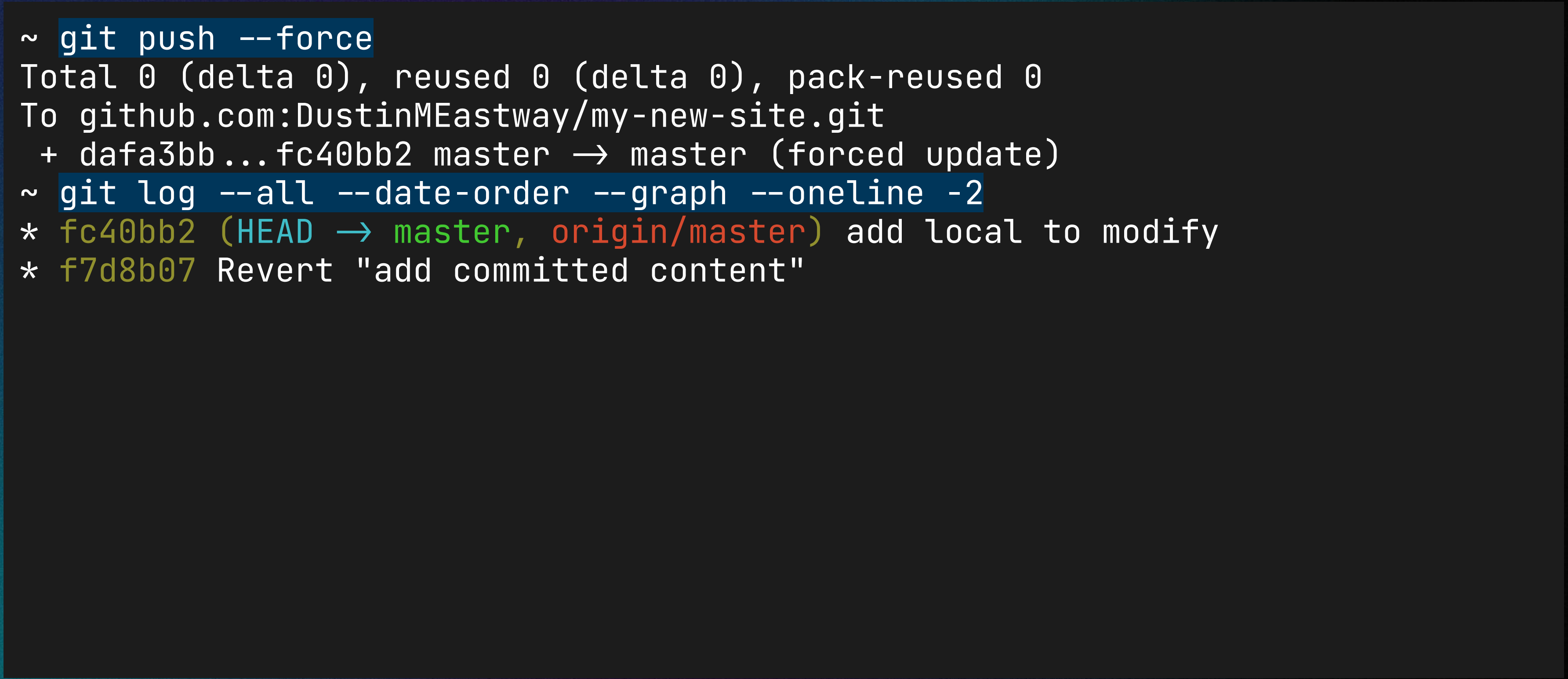


```
~ git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 782 bytes | 782.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:DustinMEastway/my-new-site.git
   f7d8b07..dafa3bb  master -> master
~ git reset --hard HEAD~
HEAD is now at fc40bb2 add local to modify
~ git log --all --date-order --graph --oneline -3
* dafa3bb (origin/master) add more local to modify
* fc40bb2 (HEAD -> master) add local to modify
* f7d8b07 Revert "add committed content"
```

How Do I Combine Histories?

Remove remote commits.

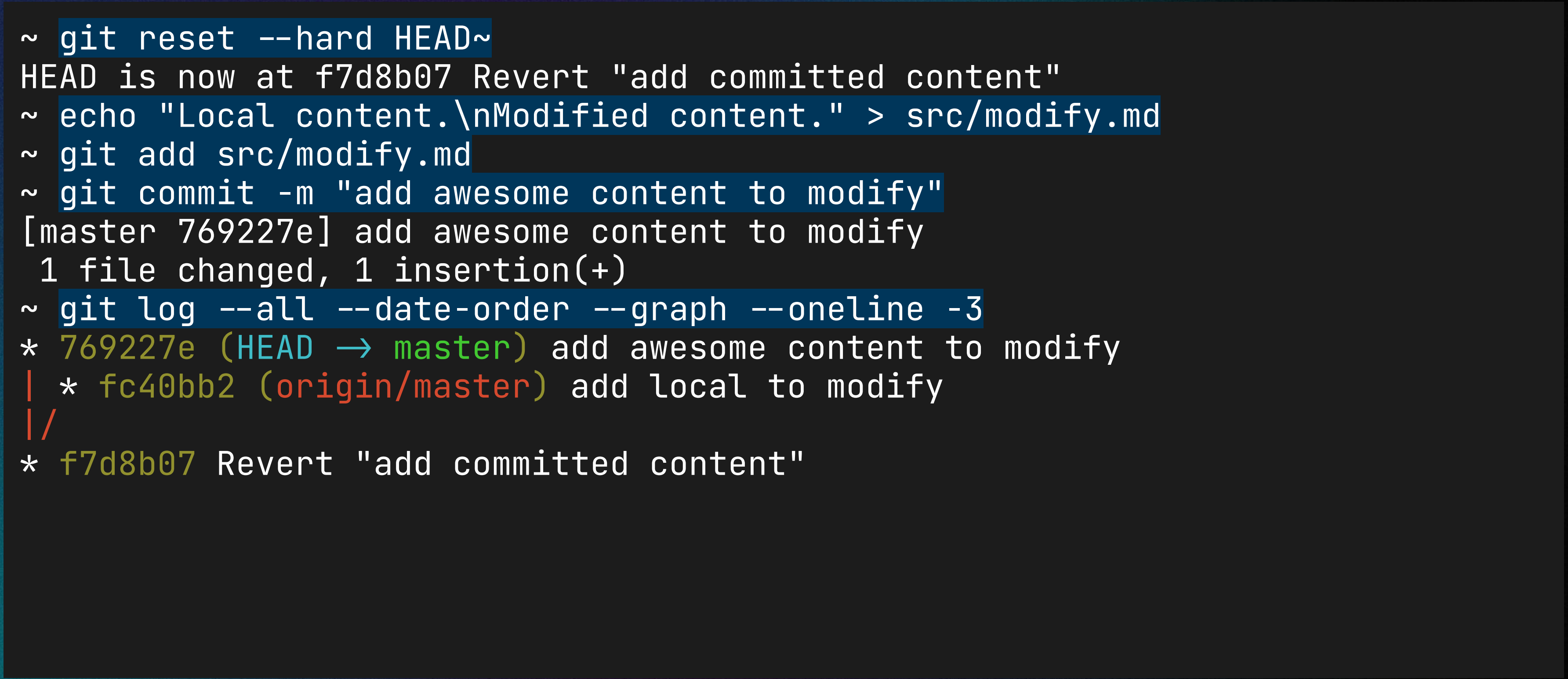
git push --force



```
~ git push --force
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:DustinMEastway/my-new-site.git
 + dafa3bb...fc40bb2 master -> master (forced update)
~ git log --all --date-order --graph --oneline -2
* fc40bb2 (HEAD -> master, origin/master) add local to modify
* f7d8b07 Revert "add committed content"
```

Prep Content

Diverge histories again...

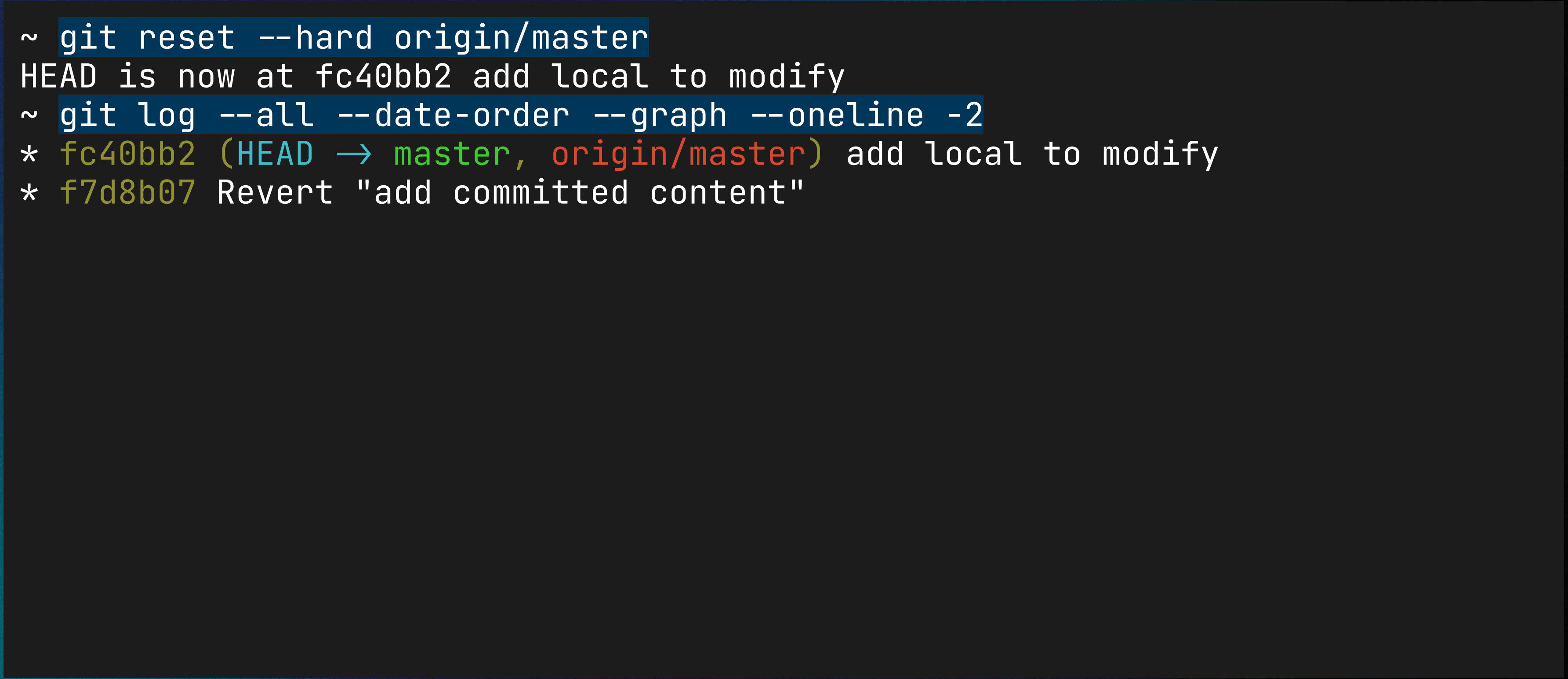


```
~ git reset --hard HEAD~
HEAD is now at f7d8b07 Revert "add committed content"
~ echo "Local content.\nModified content." > src/modify.md
~ git add src/modify.md
~ git commit -m "add awesome content to modify"
[master 769227e] add awesome content to modify
 1 file changed, 1 insertion(+)
~ git log --all --date-order --graph --oneline -3
* 769227e (HEAD -> master) add awesome content to modify
| * fc40bb2 (origin/master) add local to modify
|/
* f7d8b07 Revert "add committed content"
```

How Do I Combine Histories?

Remove local commits.

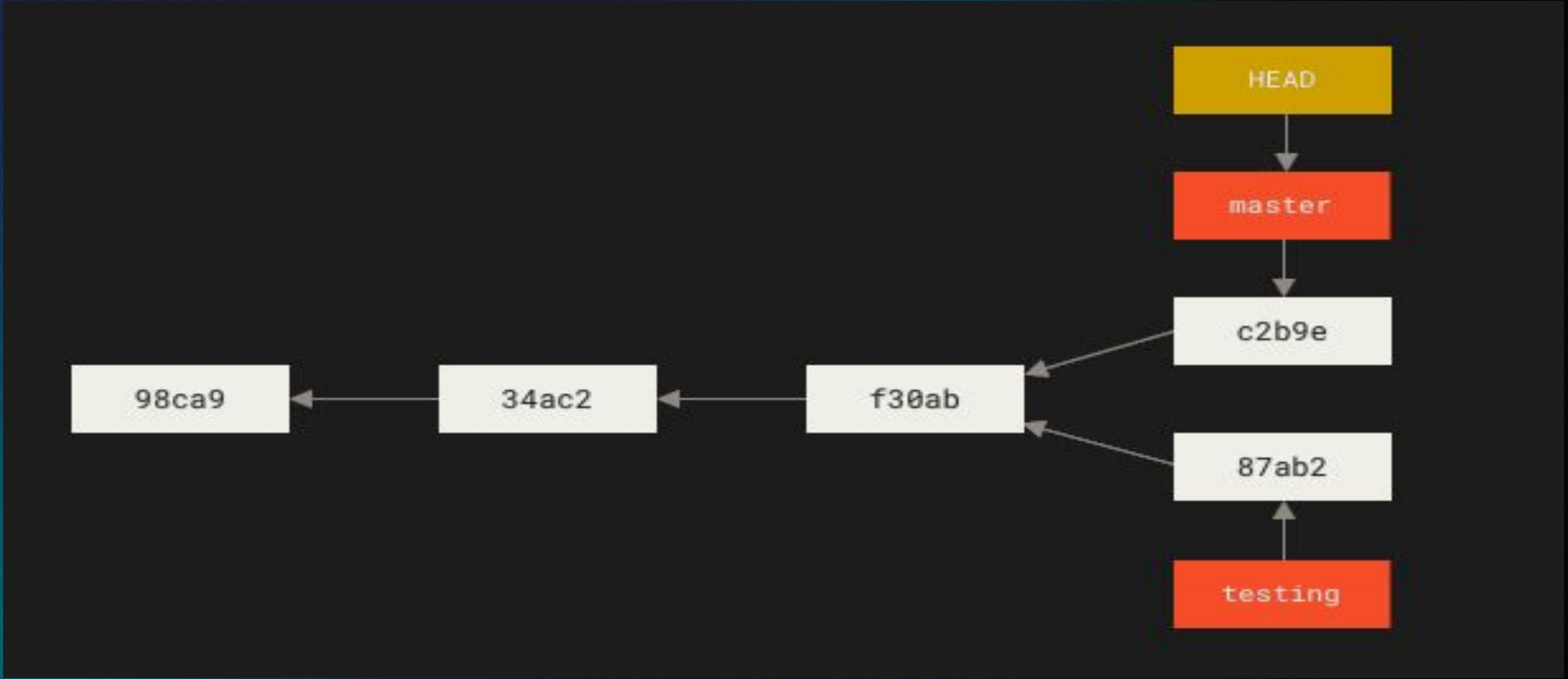
`git reset --hard <full branch name>`



```
~ git reset --hard origin/master
HEAD is now at fc40bb2 add local to modify
~ git log --all --date-order --graph --oneline -2
* fc40bb2 (HEAD → master, origin/master) add local to modify
* f7d8b07 Revert "add committed content"
```

Term: Branch

Human readable name pointing to a commit.

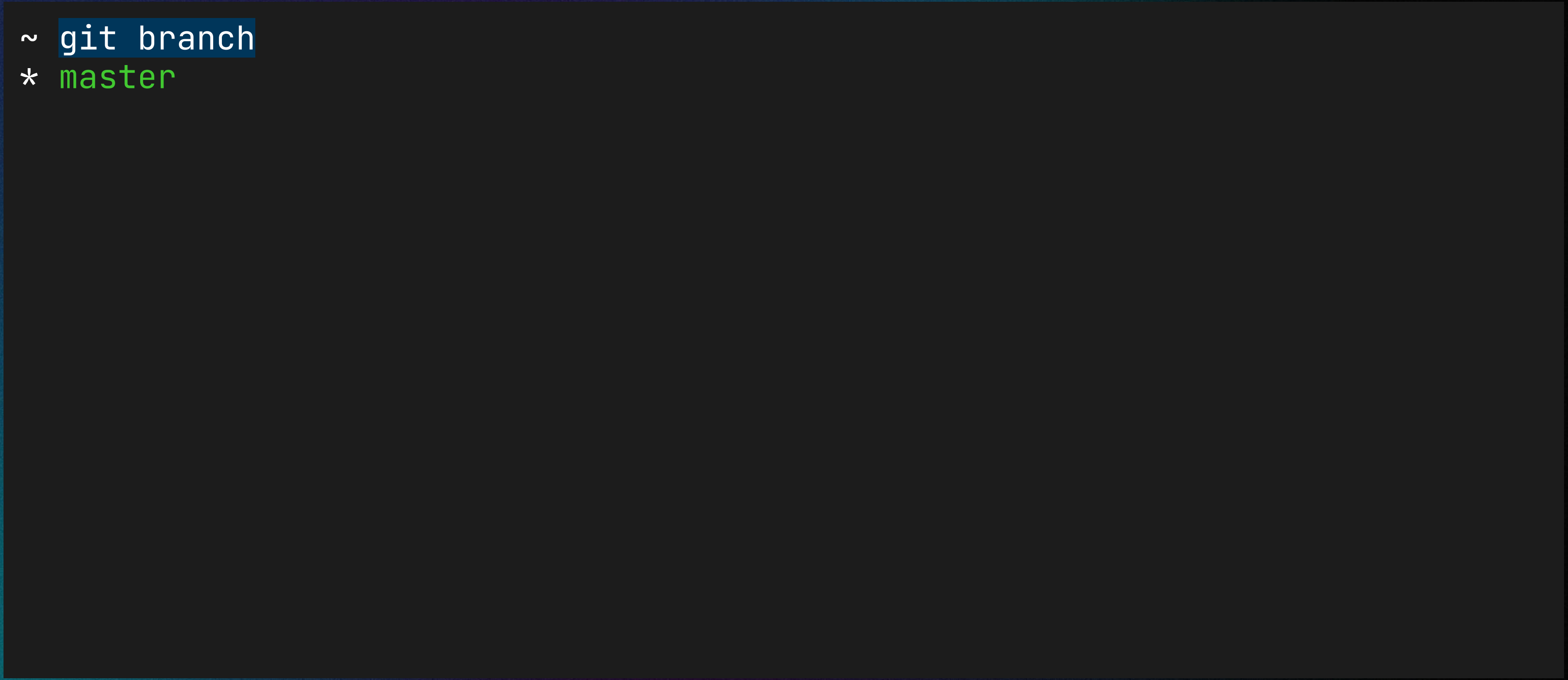


Source: Chapter 3.1 (figure 17) of the Pro Git book V2

How Do I Display Branches?

Local branches.

`git branch`

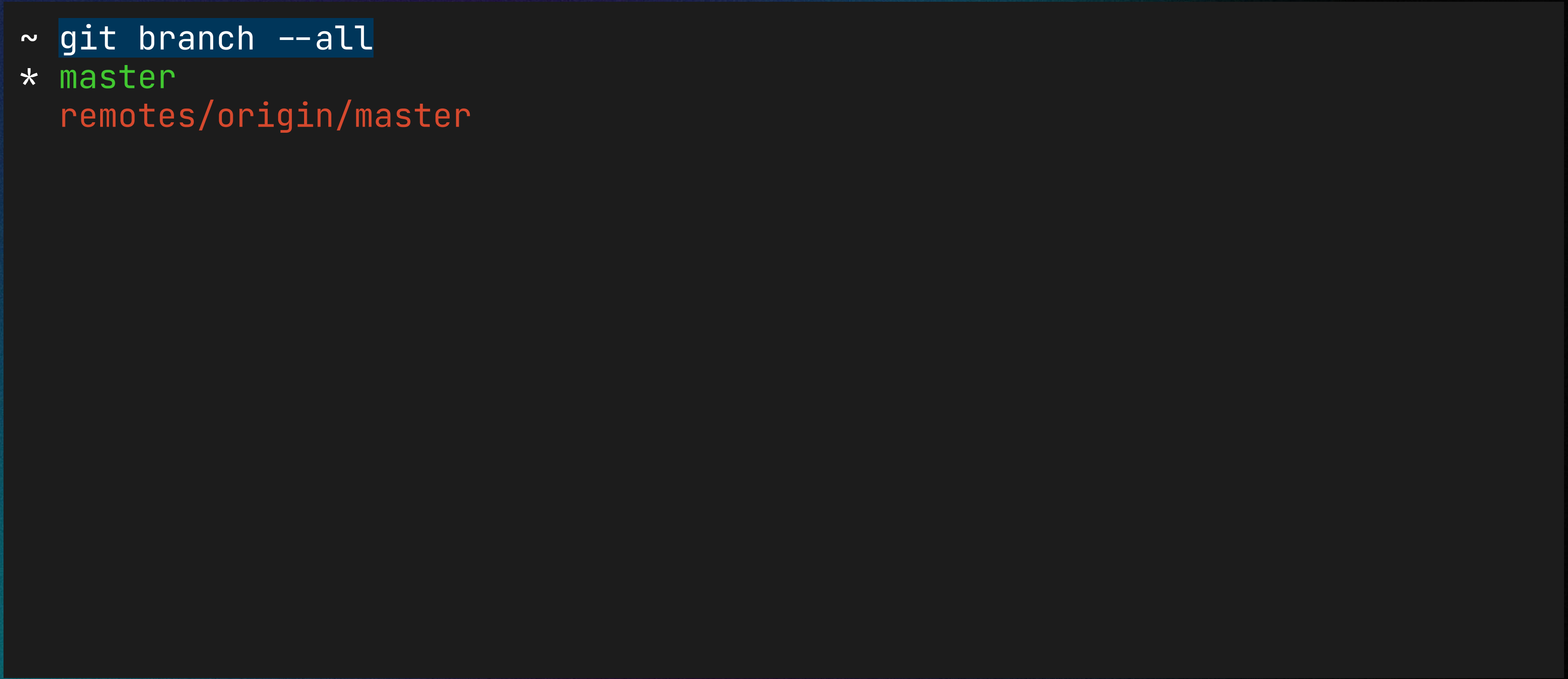


```
~ git branch
* master
```

How Do I Display Branches?

All branches.

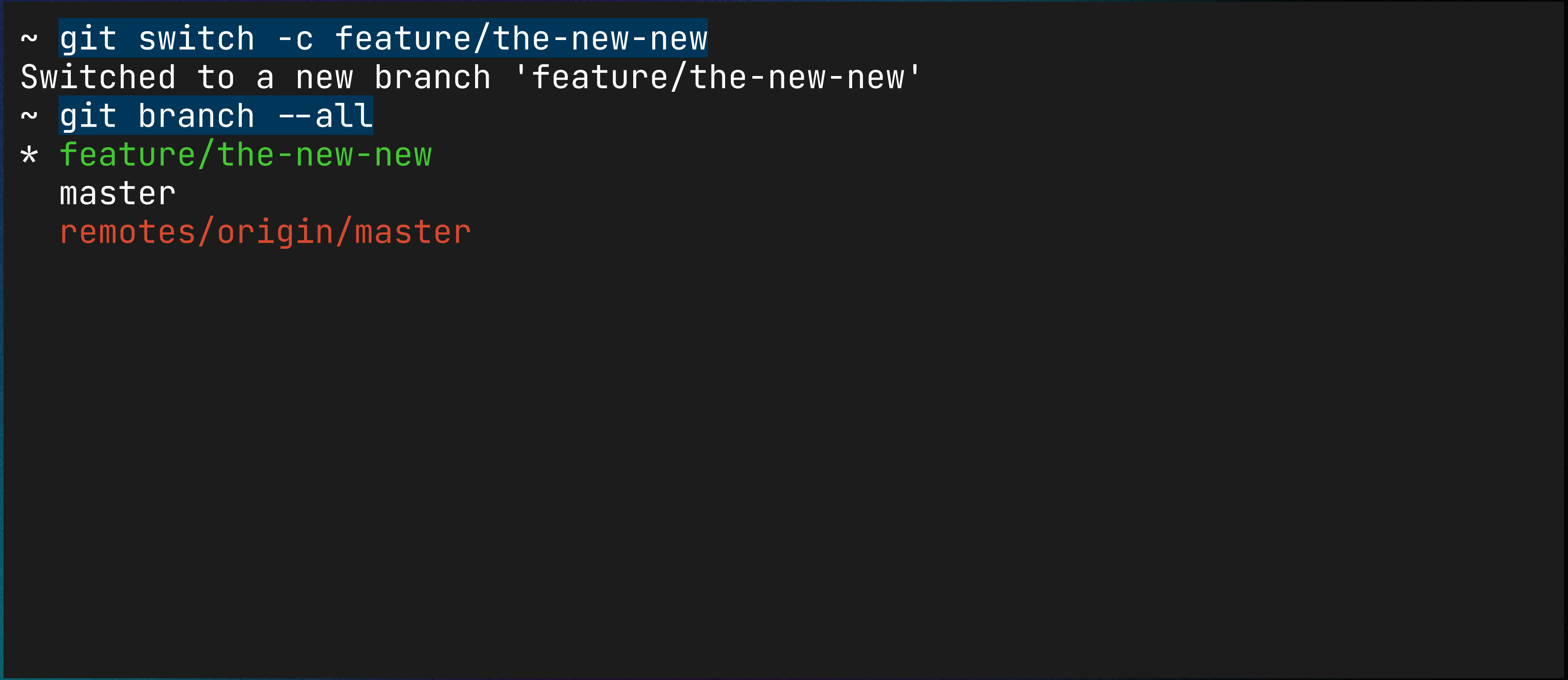
`git branch --all`



```
~ git branch --all
* master
  remotes/origin/master
```

How Do I Create A Branch?

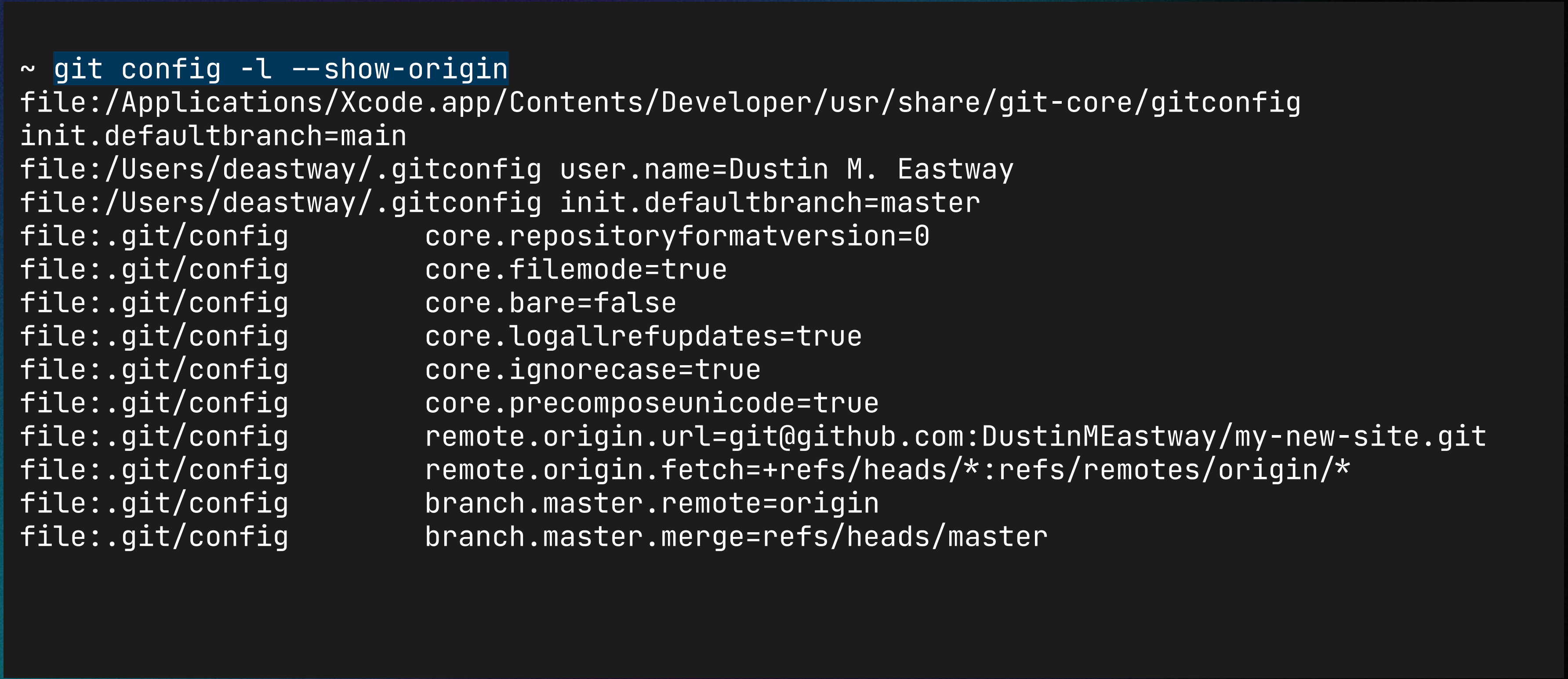
git switch -c <branch name>



```
~ git switch -c feature/the-new-new
Switched to a new branch 'feature/the-new-new'
~ git branch --all
* feature/the-new-new
  master
  remotes/origin/master
```

How Do I Display Git Configuration?

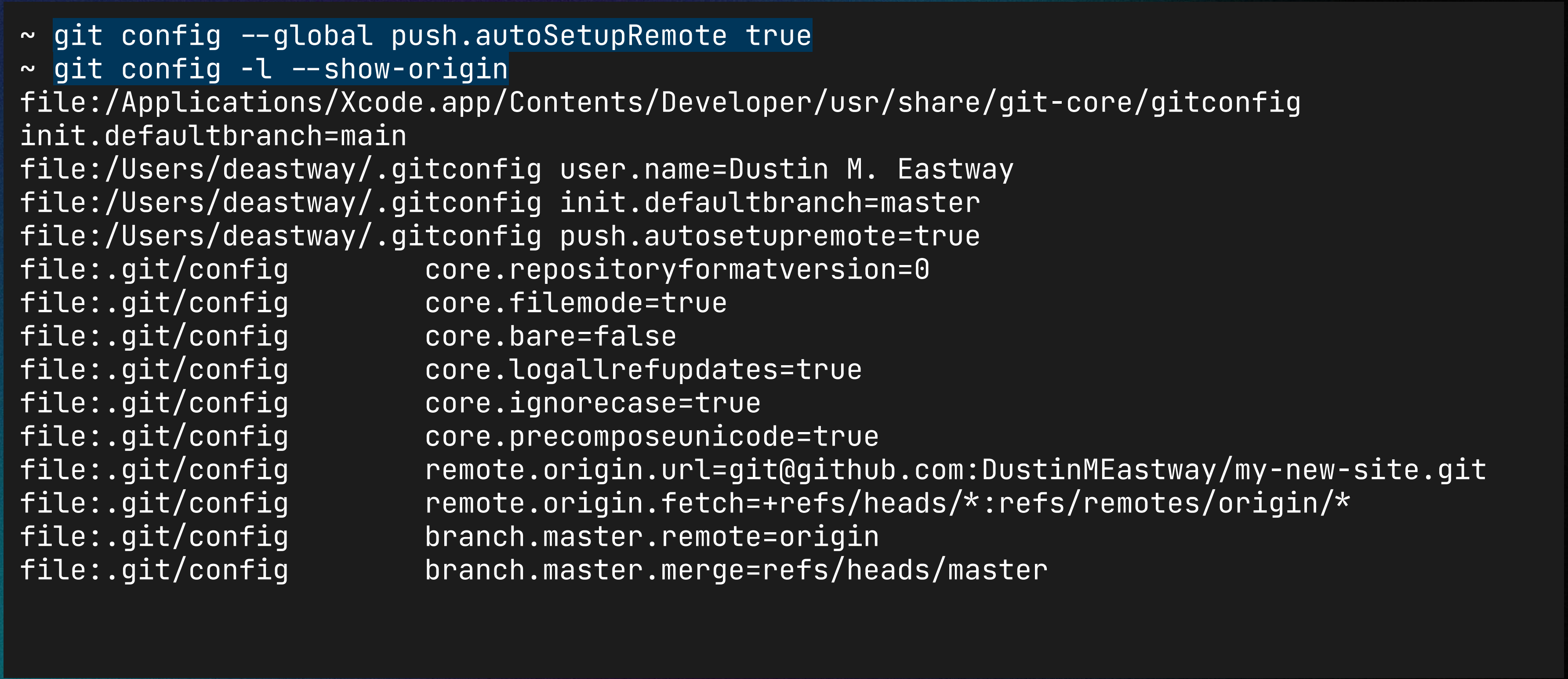
git config -l [--show-origin]



```
~ git config -l --show-origin
file:/Applications/Xcode.app/Contents/Developer/usr/share/git-core/gitconfig
init.defaultbranch=main
file:/Users/deastway/.gitconfig user.name=Dustin M. Eastway
file:/Users/deastway/.gitconfig init.defaultbranch=master
file:.git/config        core.repositoryformatversion=0
file:.git/config        core.filemode=true
file:.git/config        core.bare=false
file:.git/config        core.logallrefupdates=true
file:.git/config        core.ignorecase=true
file:.git/config        core.precomposeunicode=true
file:.git/config        remote.origin.url=git@github.com:DustinMEastway/my-new-site.git
file:.git/config        remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
file:.git/config        branch.master.remote=origin
file:.git/config        branch.master.merge=refs/heads/master
```

How Do I Set Git Configuration?

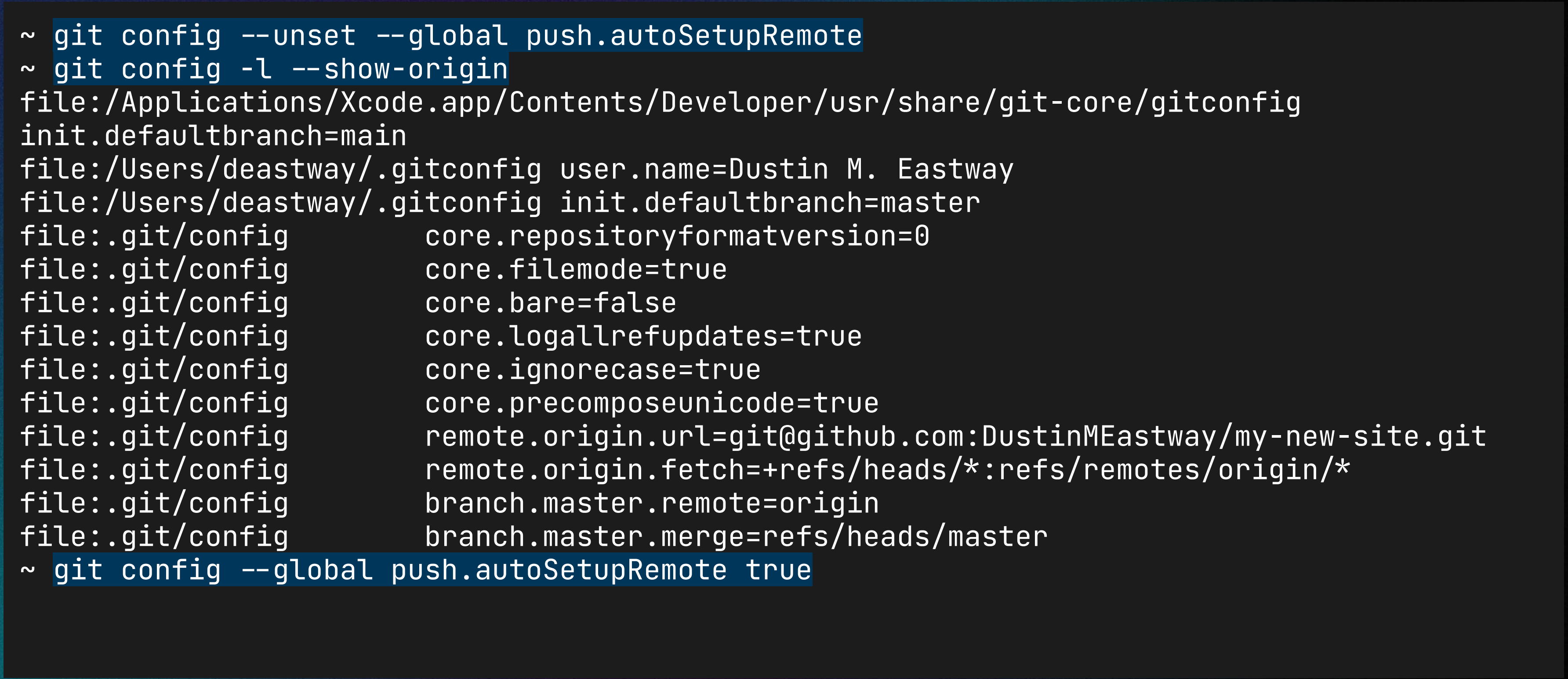
git config [--global] <config key> <config value>



```
~ git config --global push.autoSetupRemote true
~ git config -l --show-origin
file:/Applications/Xcode.app/Contents/Developer/usr/share/git-core/gitconfig
init.defaultbranch=main
file:/Users/deastway/.gitconfig user.name=Dustin M. Eastway
file:/Users/deastway/.gitconfig init.defaultbranch=master
file:/Users/deastway/.gitconfig push.autosetupremote=true
file:.git/config        core.repositoryformatversion=0
file:.git/config        core.filemode=true
file:.git/config        core.bare=false
file:.git/config        core.logallrefupdates=true
file:.git/config        core.ignorecase=true
file:.git/config        core.precomposeunicode=true
file:.git/config        remote.origin.url=git@github.com:DustinMEastway/my-new-site.git
file:.git/config        remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
file:.git/config        branch.master.remote=origin
file:.git/config        branch.master.merge=refs/heads/master
```

How Do I Remove Git Configuration?

git config --unset [--global] <config key>

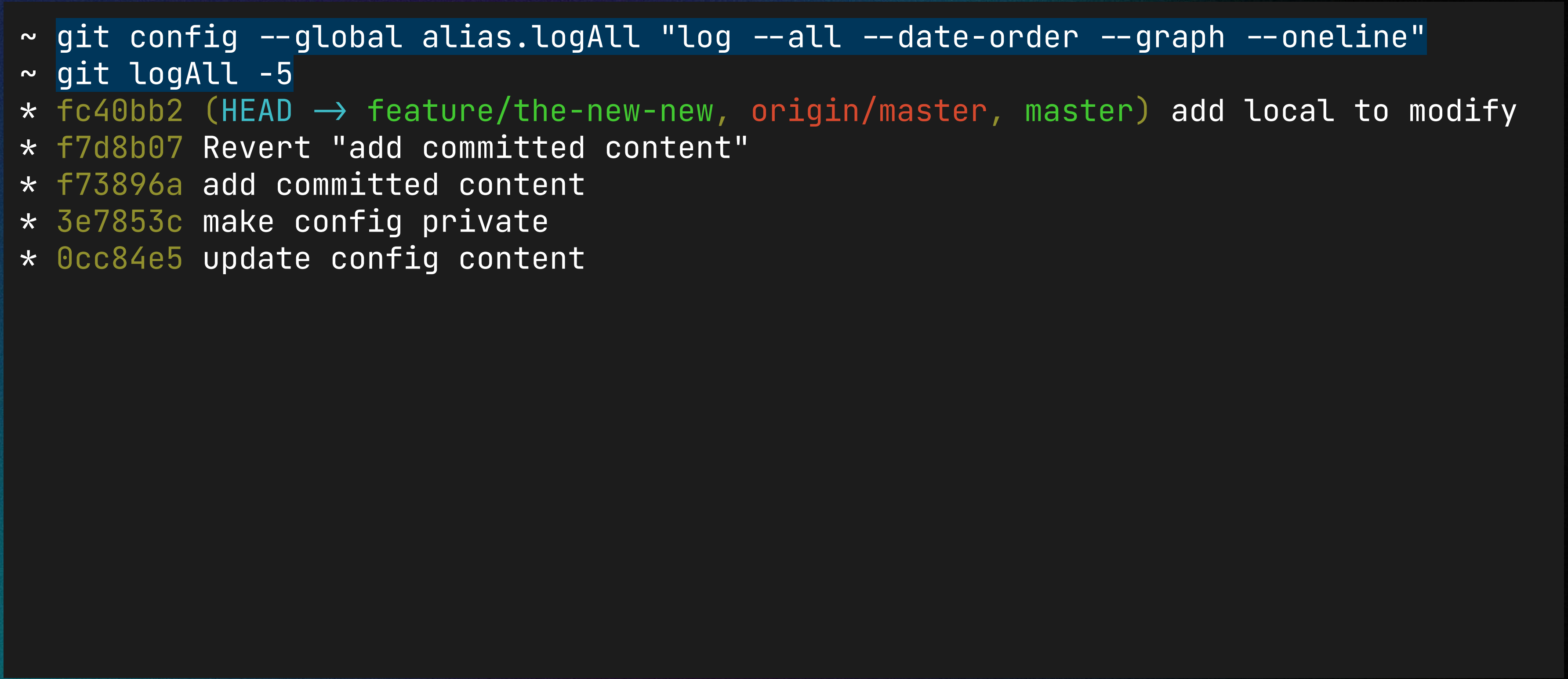


```
~ git config --unset --global push.autoSetupRemote
~ git config -l --show-origin
file:/Applications/Xcode.app/Contents/Developer/usr/share/git-core/gitconfig
init.defaultbranch=main
file:/Users/deastway/.gitconfig user.name=Dustin M. Eastway
file:/Users/deastway/.gitconfig init.defaultbranch=master
file:.git/config        core.repositoryformatversion=0
file:.git/config        core.filemode=true
file:.git/config        core.bare=false
file:.git/config        core.logallrefupdates=true
file:.git/config        core.ignorecase=true
file:.git/config        core.precomposeunicode=true
file:.git/config        remote.origin.url=git@github.com:DustinMEastway/my-new-site.git
file:.git/config        remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
file:.git/config        branch.master.remote=origin
file:.git/config        branch.master.merge=refs/heads/master
~ git config --global push.autoSetupRemote true
```

How Do I Alias Commands?

Internal commands.

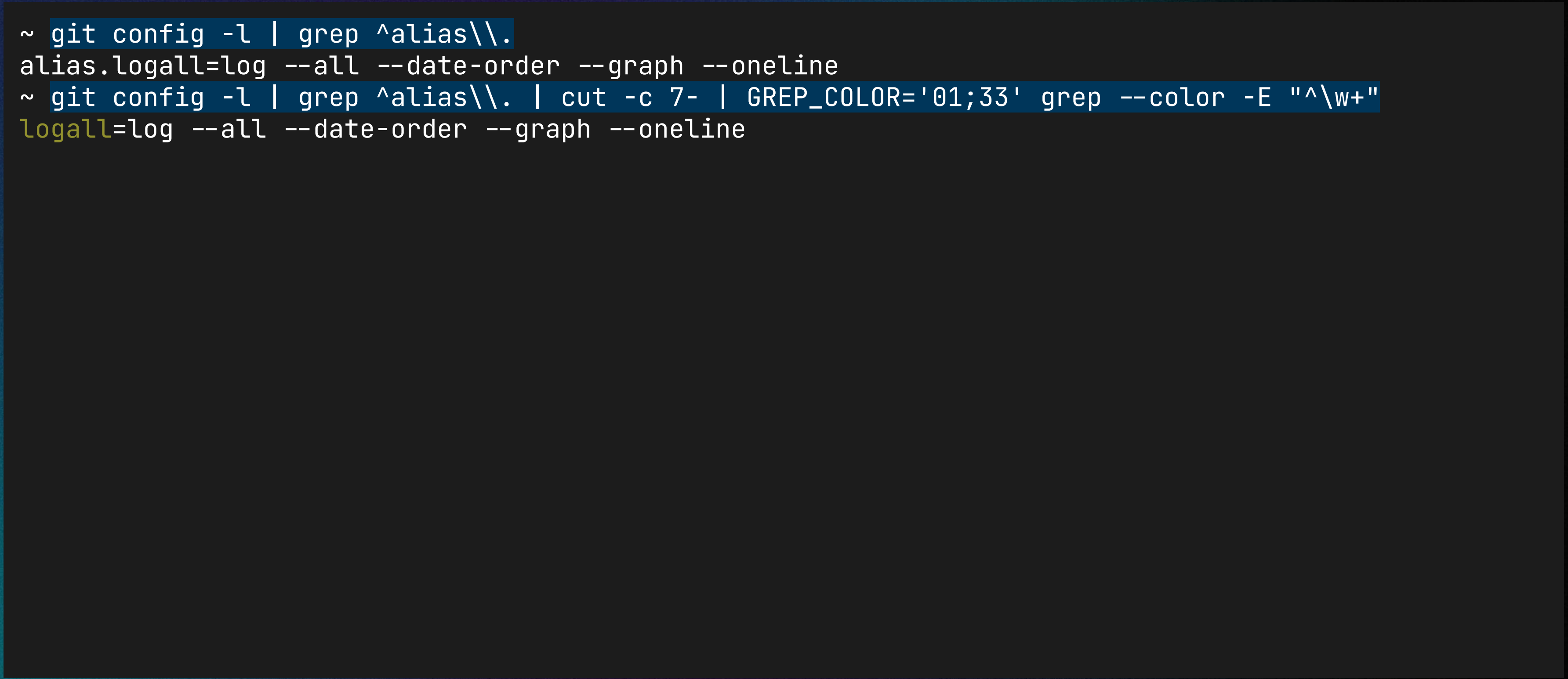
git config [--global] alias.<alias name> "<git command>"



```
~ git config --global alias.logAll "log --all --date-order --graph --oneline"
~ git logAll -5
* fc40bb2 (HEAD -> feature/the-new-new, origin/master, master) add local to modify
* f7d8b07 Revert "add committed content"
* f73896a add committed content
* 3e7853c make config private
* 0cc84e5 update config content
```

How Do I Display Aliases?

git config -l | grep ^alias\\.

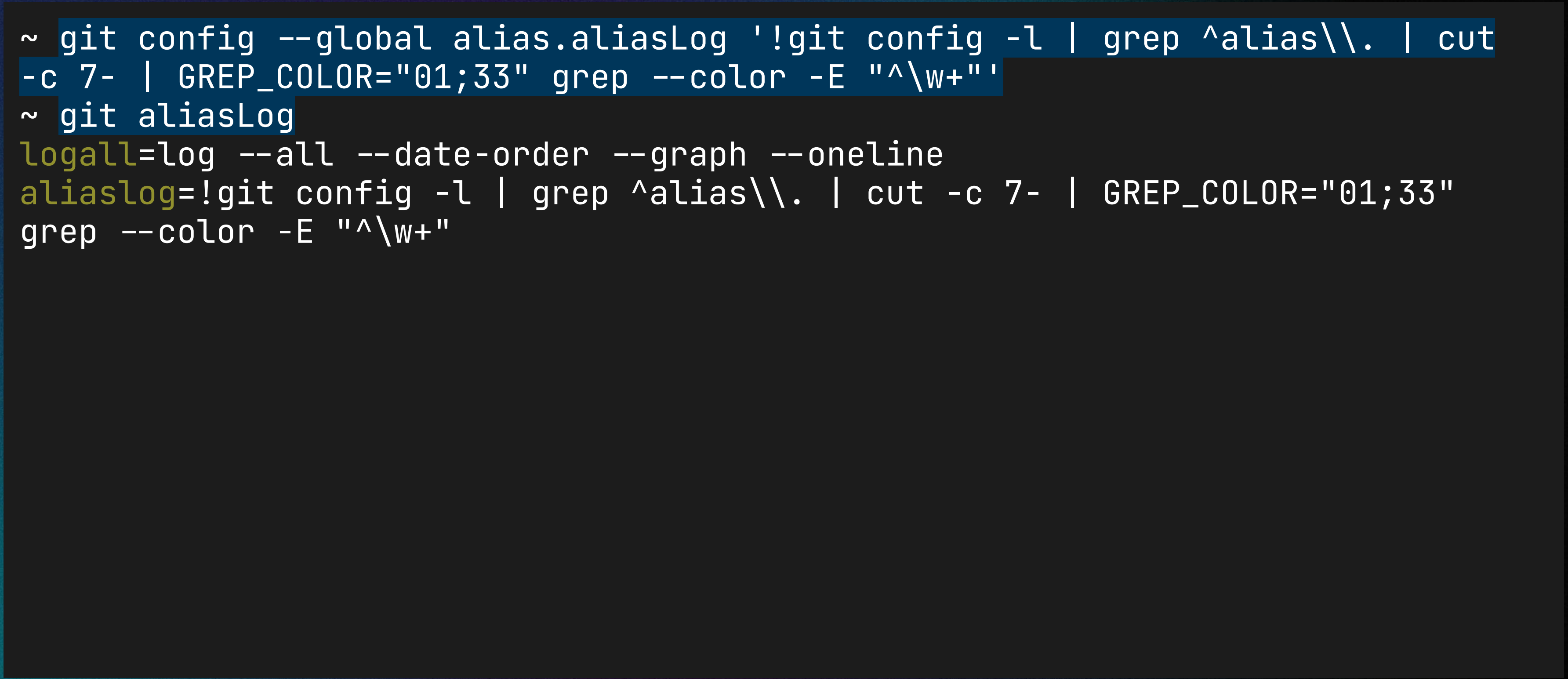


```
~ git config -l | grep ^alias\\.
alias.logall=log --all --date-order --graph --oneline
~ git config -l | grep ^alias\\. | cut -c 7- | GREP_COLOR='01;33' grep --color -E "^\w+"
logall=log --all --date-order --graph --oneline
```

How Do I Alias Commands?

Any commands.

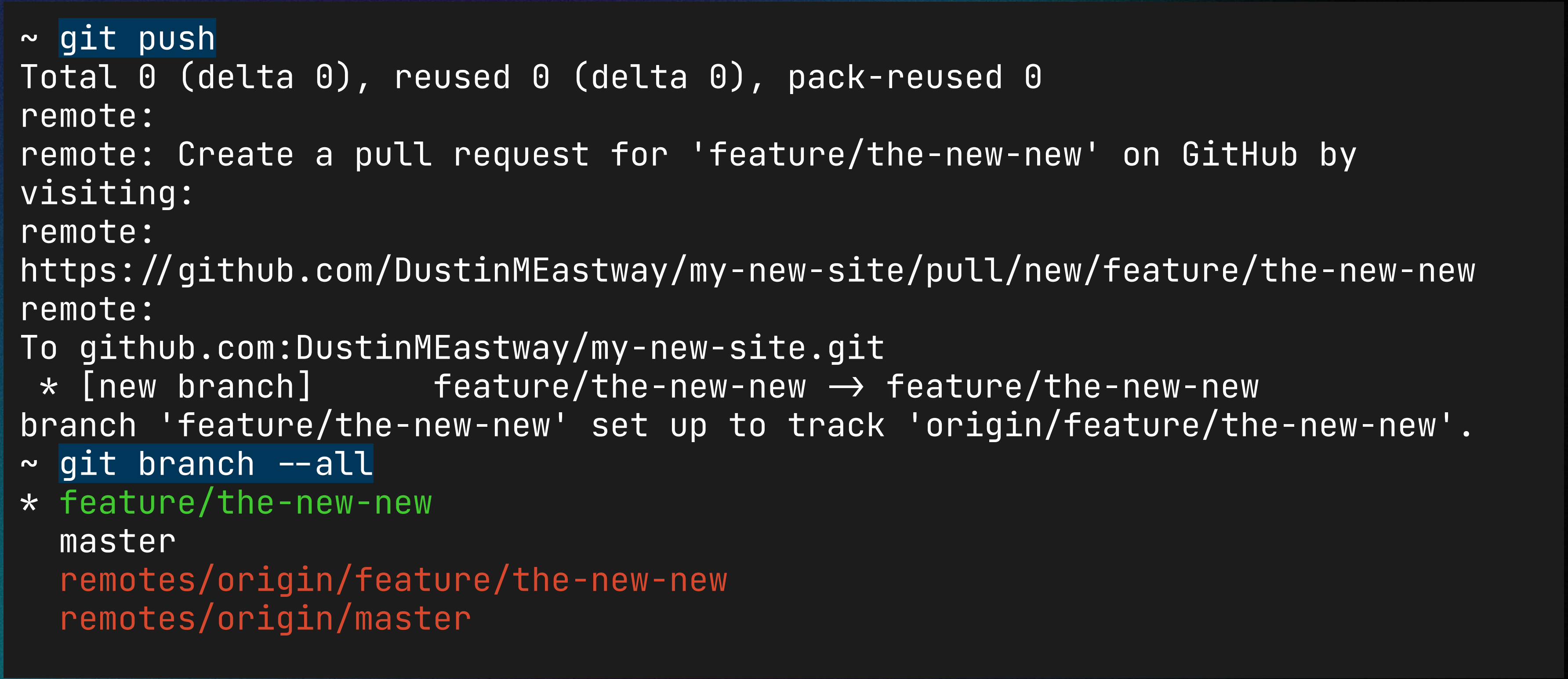
git config [--global] alias.<alias name> '!<any full command>'



```
~ git config --global alias.aliasLog '!git config -l | grep ^alias\\. | cut
-c 7- | GREP_COLOR="01;33" grep --color -E "^\w+"'
~ git aliasLog
logall=log --all --date-order --graph --oneline
aliaslog=!git config -l | grep ^alias\\. | cut -c 7- | GREP_COLOR="01;33"
grep --color -E "^\w+"
```

How Do I Upload A Branch?

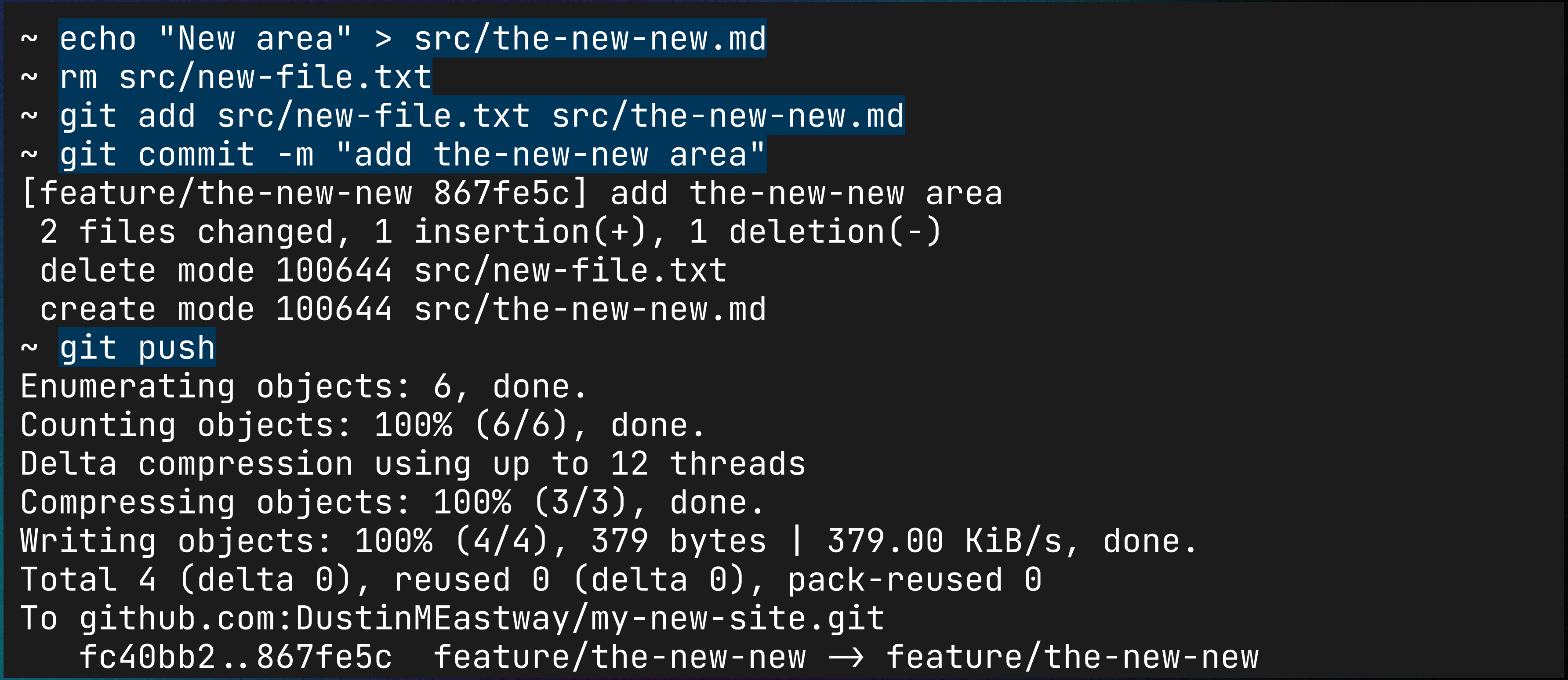
git push



```
~ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/the-new-new' on GitHub by
visiting:
remote:
https://github.com/DustinMEastway/my-new-site/pull/new/feature/the-new-new
remote:
To github.com:DustinMEastway/my-new-site.git
 * [new branch]      feature/the-new-new -> feature/the-new-new
branch 'feature/the-new-new' set up to track 'origin/feature/the-new-new'.
~ git branch --all
* feature/the-new-new
  master
  remotes/origin/feature/the-new-new
  remotes/origin/master
```

Prep Content

Add commit to non-master branch.



```
~ echo "New area" > src/the-new-new.md
~ rm src/new-file.txt
~ git add src/new-file.txt src/the-new-new.md
~ git commit -m "add the-new-new area"
[feature/the-new-new 867fe5c] add the-new-new area
 2 files changed, 1 insertion(+), 1 deletion(-)
 delete mode 100644 src/new-file.txt
 create mode 100644 src/the-new-new.md
~ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 379 bytes | 379.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:DustinMEastway/my-new-site.git
   fc40bb2..867fe5c  feature/the-new-new → feature/the-new-new
```

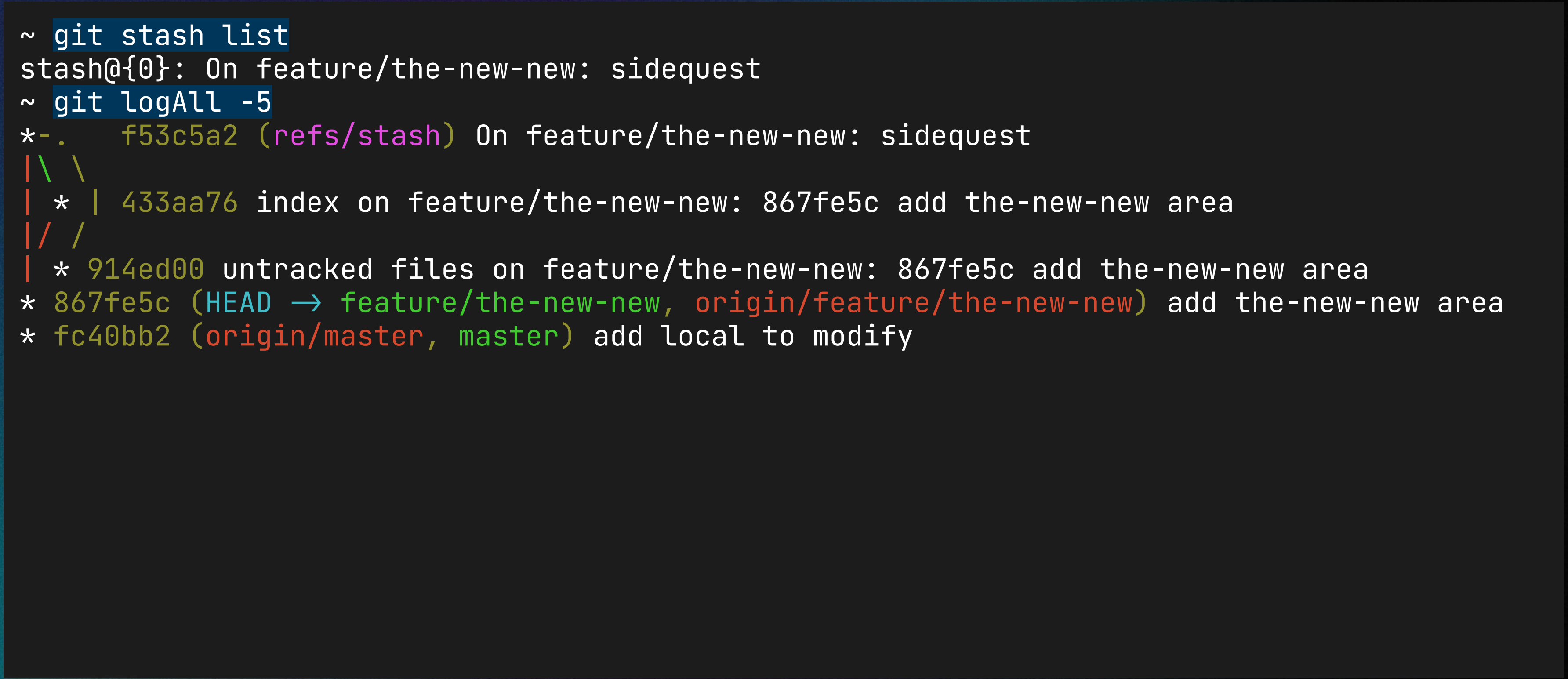
How Do I Store Uncommitted Changes?

git stash [save]

```
~ echo "Content to stash." > src/stashable.md
~ git status
On branch feature/the-new-new
Your branch is up to date with 'origin/feature/the-new-new'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  src/stashable.md

nothing added to commit but untracked files present (use "git add" to track)
~ git stash -u -m "sidequest"
Saved working directory and index state On feature/the-new-new: sidequest
~ git status
On branch feature/the-new-new
Your branch is up to date with 'origin/feature/the-new-new'.

nothing to commit, working tree clean
```

How Do I List Stored Changes?

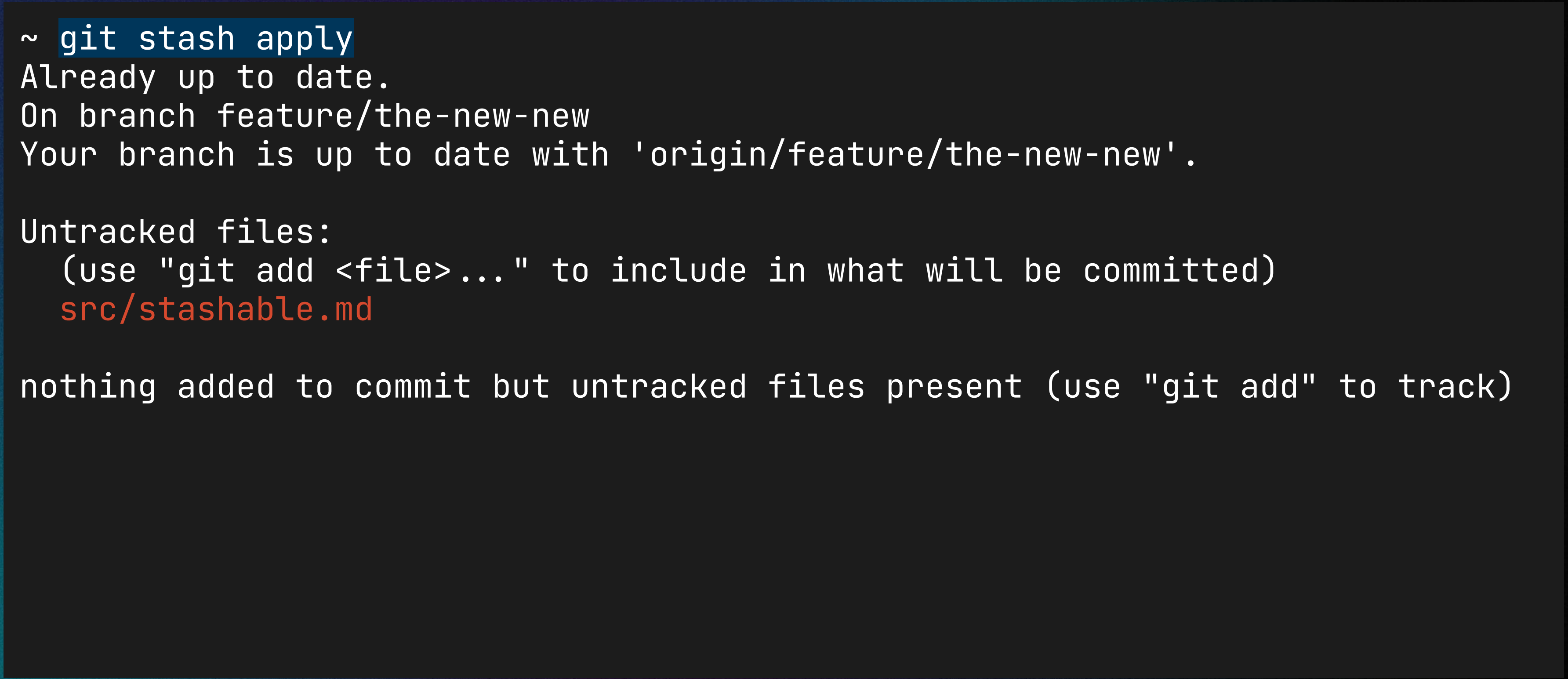
git stash list



```
~ git stash list
stash@{0}: On feature/the-new-new: sidequest
~ git logAll -5
*-.   f53c5a2 (refs/stash) On feature/the-new-new: sidequest
|\ \
| * | 433aa76 index on feature/the-new-new: 867fe5c add the-new-new area
|/ /
| * 914ed00 untracked files on feature/the-new-new: 867fe5c add the-new-new area
* 867fe5c (HEAD -> feature/the-new-new, origin/feature/the-new-new) add the-new-new area
* fc40bb2 (origin/master, master) add local to modify
```

How Do I Get Stored Changes?

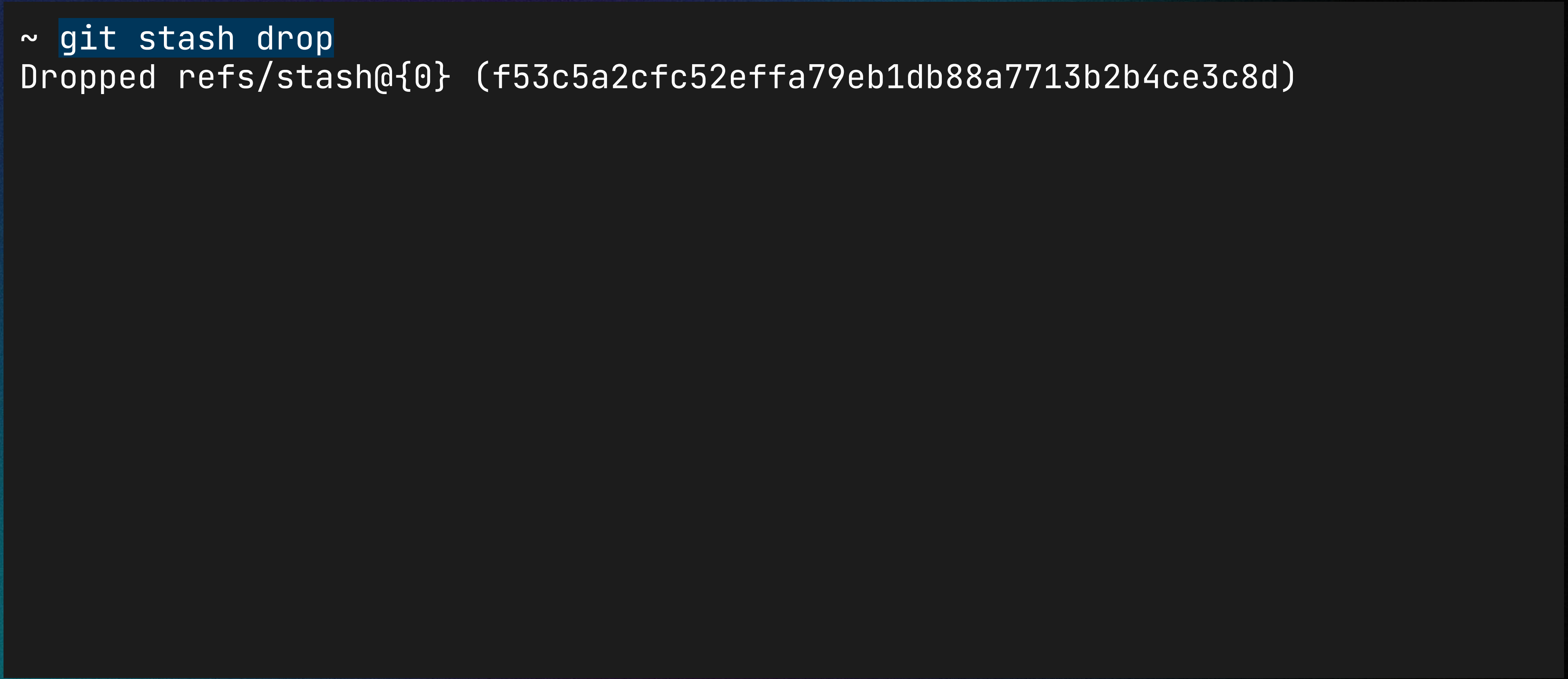
`git stash apply [stash@{<stash index>}]`



```
~ git stash apply
Already up to date.
On branch feature/the-new-new
Your branch is up to date with 'origin/feature/the-new-new'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  src/stashable.md

nothing added to commit but untracked files present (use "git add" to track)
```

How Do I Delete Stored Changes?

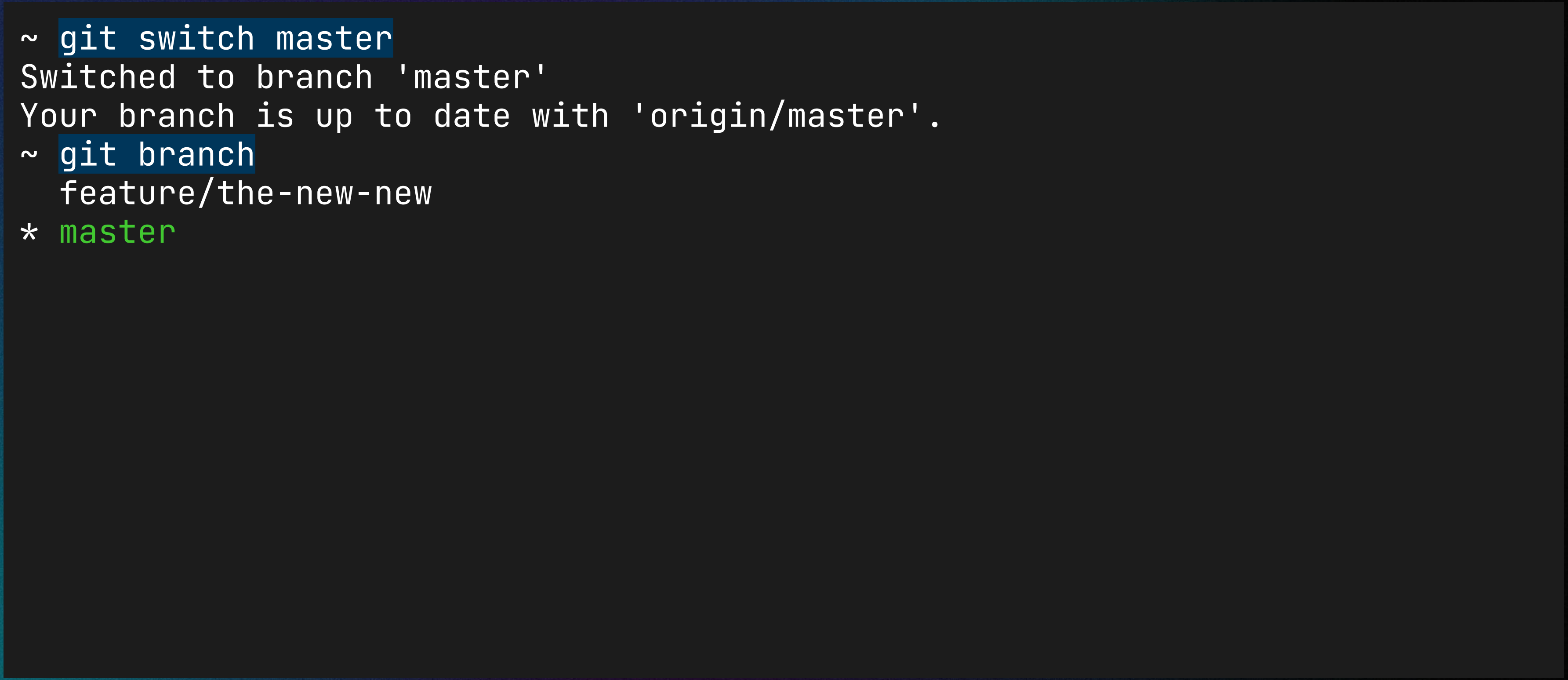
`git stash drop [stash@{<stash index>}]`



```
~ git stash drop
Dropped refs/stash@{0} (f53c5a2cfc52effa79eb1db88a7713b2b4ce3c8d)
```

How Do I Switch Branches?

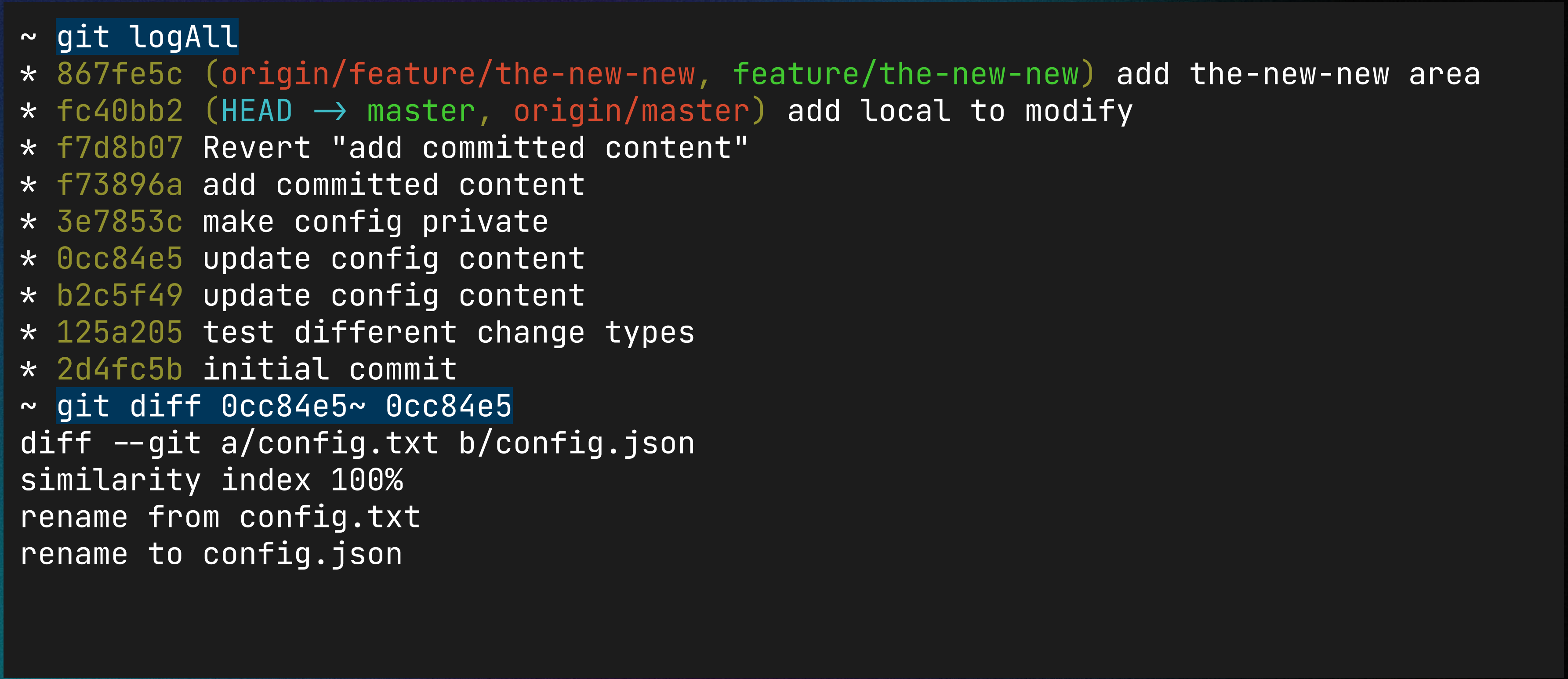
git switch <branch name>



```
~ git switch master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
~ git branch
  feature/the-new-new
* master
```

Prep Content

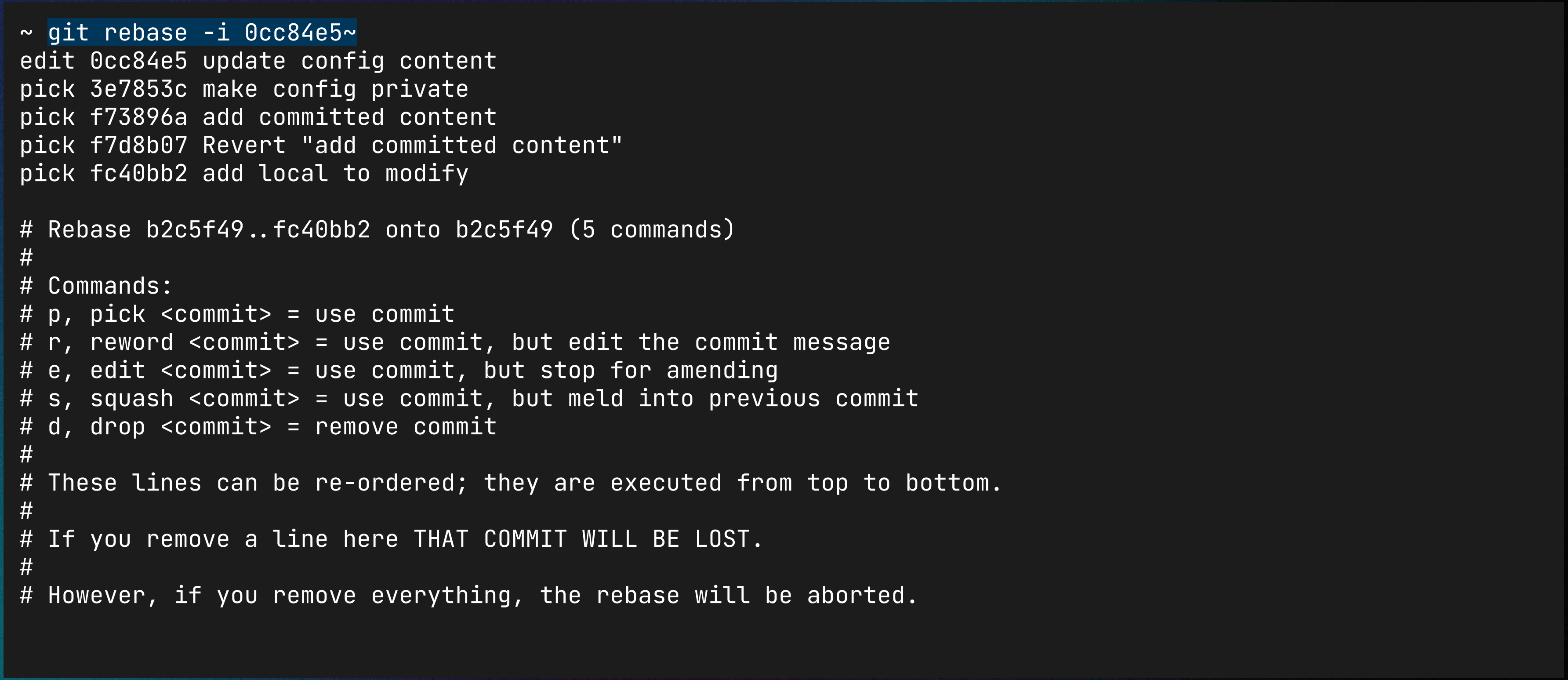
Display multiple branches.



```
~ git logAll
* 867fe5c (origin/feature/the-new-new, feature/the-new-new) add the-new-new area
* fc40bb2 (HEAD → master, origin/master) add local to modify
* f7d8b07 Revert "add committed content"
* f73896a add committed content
* 3e7853c make config private
* 0cc84e5 update config content
* b2c5f49 update config content
* 125a205 test different change types
* 2d4fc5b initial commit
~ git diff 0cc84e5~ 0cc84e5
diff --git a/config.txt b/config.json
similarity index 100%
rename from config.txt
rename to config.json
```

How Do I “Modify A Commit”?

git rebase -i <commit hash to modify>~

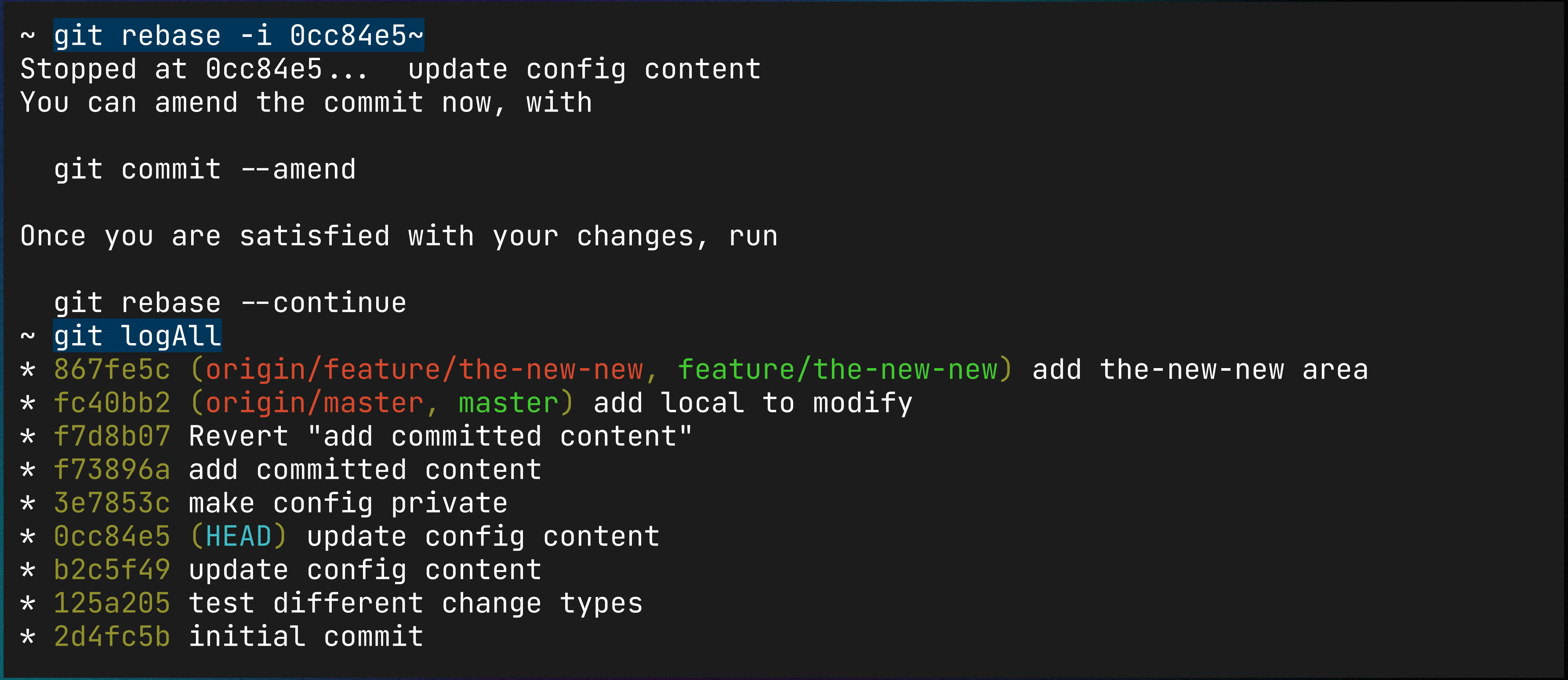


```
~ git rebase -i 0cc84e5~
edit 0cc84e5 update config content
pick 3e7853c make config private
pick f73896a add committed content
pick f7d8b07 Revert "add committed content"
pick fc40bb2 add local to modify

# Rebase b2c5f49..fc40bb2 onto b2c5f49 (5 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# d, drop <commit> = remove commit
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
```

How Do I “Modify A Commit”?

Continued...

git rebase -i <commit hash to modify>~

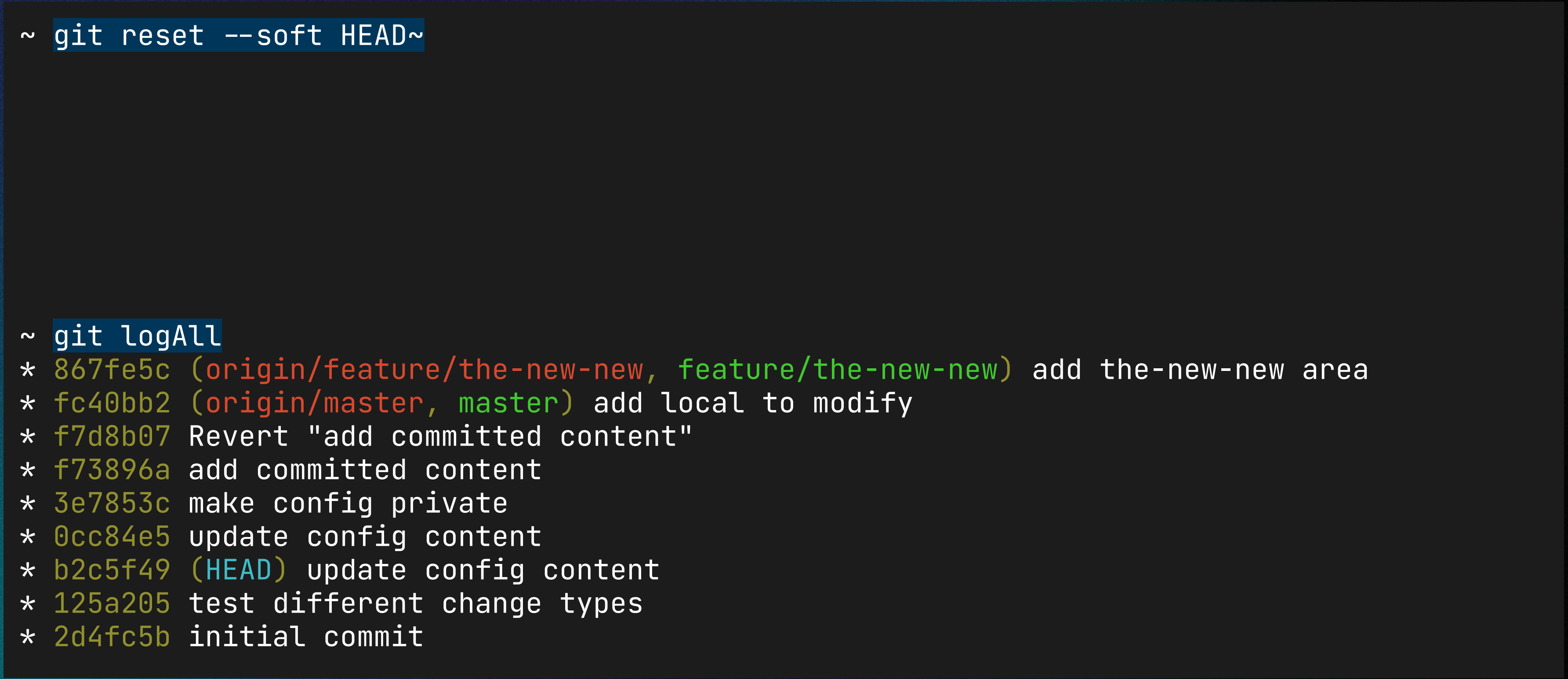


```
~ git rebase -i 0cc84e5~
Stopped at 0cc84e5...  update config content
You can amend the commit now, with

  git commit --amend

Once you are satisfied with your changes, run

  git rebase --continue
~ git logAll
* 867fe5c (origin/feature/the-new-new, feature/the-new-new) add the-new-new area
* fc40bb2 (origin/master, master) add local to modify
* f7d8b07 Revert "add committed content"
* f73896a add committed content
* 3e7853c make config private
* 0cc84e5 (HEAD) update config content
* b2c5f49 update config content
* 125a205 test different change types
* 2d4fc5b initial commit
```

How Do I “Modify A Commit”?

Continued...

git reset --soft HEAD~

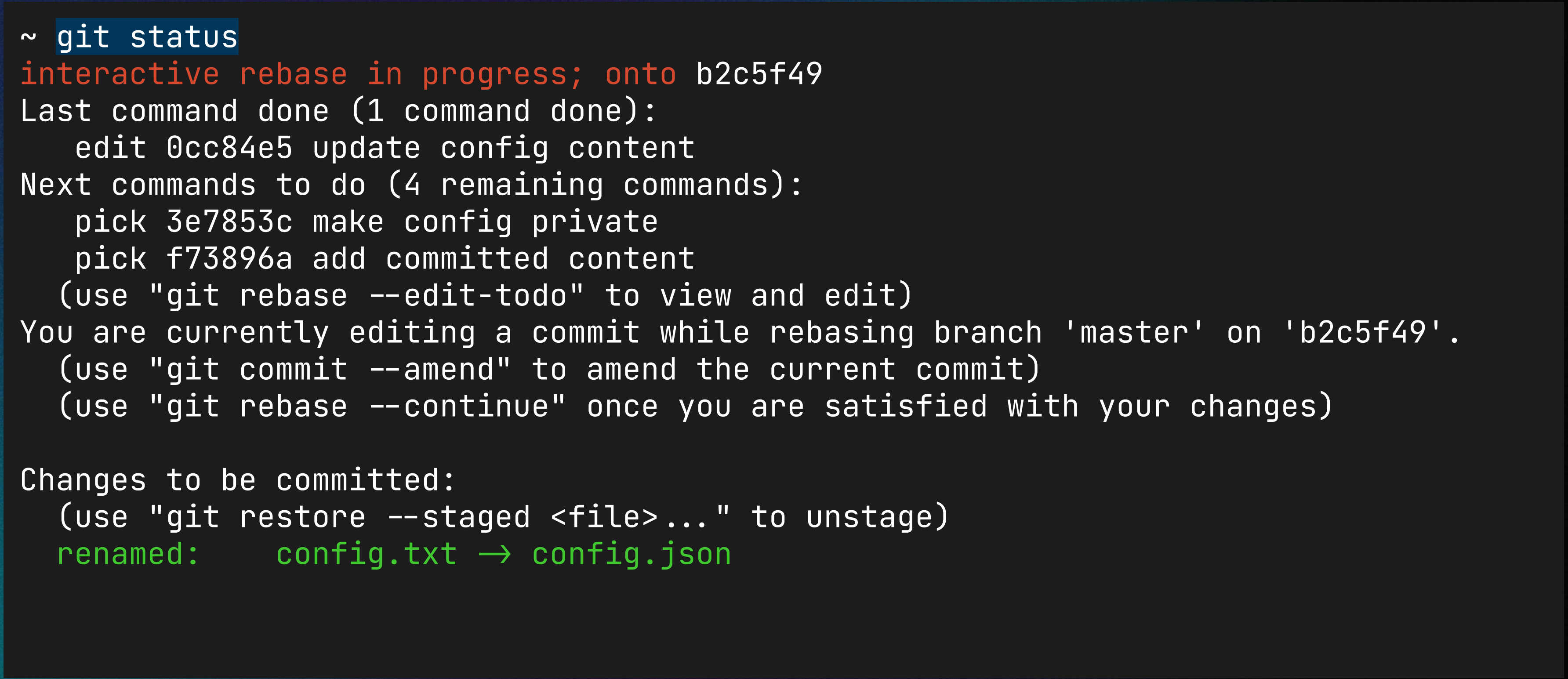


```
~ git reset --soft HEAD~








~ git logAll
* 867fe5c (origin/feature/the-new-new, feature/the-new-new) add the-new-new area
* fc40bb2 (origin/master, master) add local to modify
* f7d8b07 Revert "add committed content"
* f73896a add committed content
* 3e7853c make config private
* 0cc84e5 update config content
* b2c5f49 (HEAD) update config content
* 125a205 test different change types
* 2d4fc5b initial commit
```

How Do I “Modify A Commit”?

Continued...

`git reset --soft HEAD~`

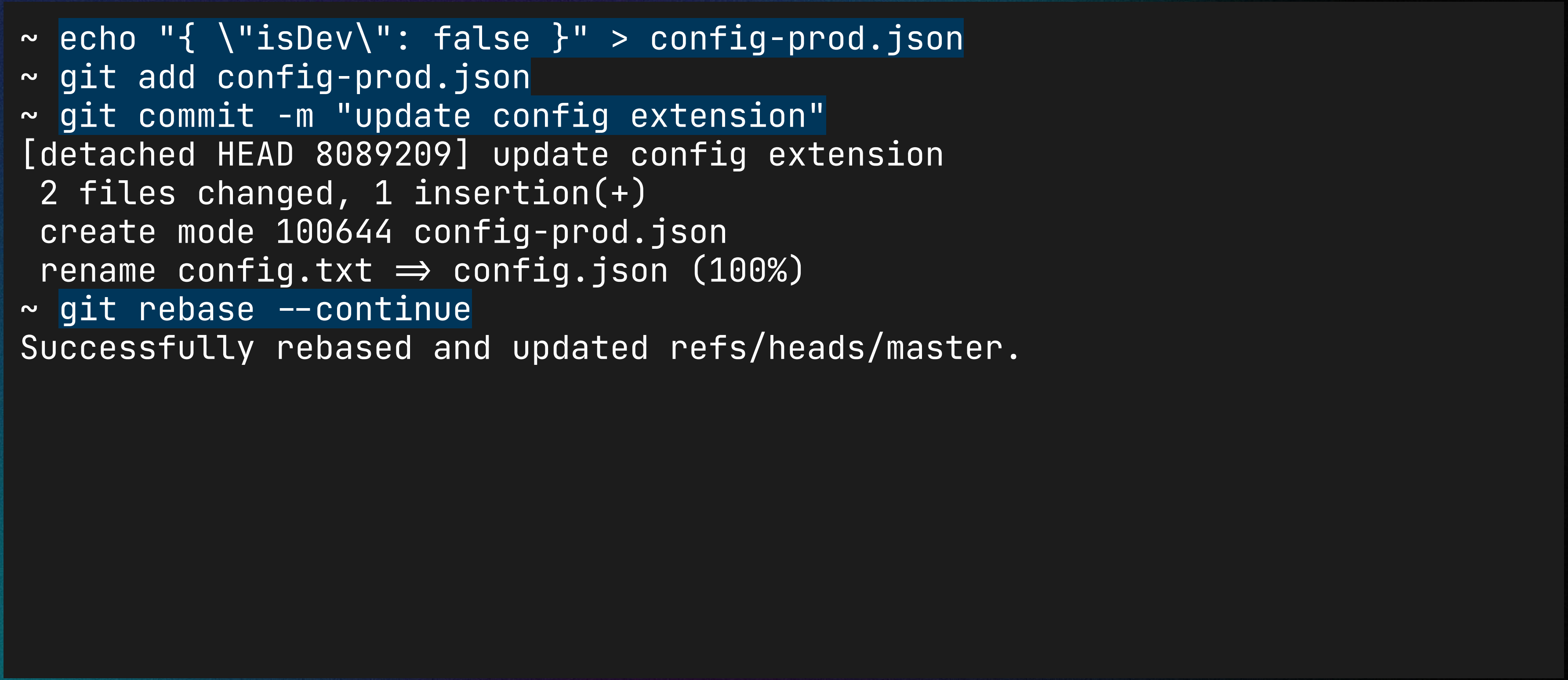


```
~ git status
interactive rebase in progress; onto b2c5f49
Last command done (1 command done):
   edit 0cc84e5 update config content
Next commands to do (4 remaining commands):
   pick 3e7853c make config private
   pick f73896a add committed content
  (use "git rebase --edit-todo" to view and edit)
You are currently editing a commit while rebasing branch 'master' on 'b2c5f49'.
  (use "git commit --amend" to amend the current commit)
  (use "git rebase --continue" once you are satisfied with your changes)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  renamed:    config.txt -> config.json
```

How Do I “Modify A Commit”?

Continued...

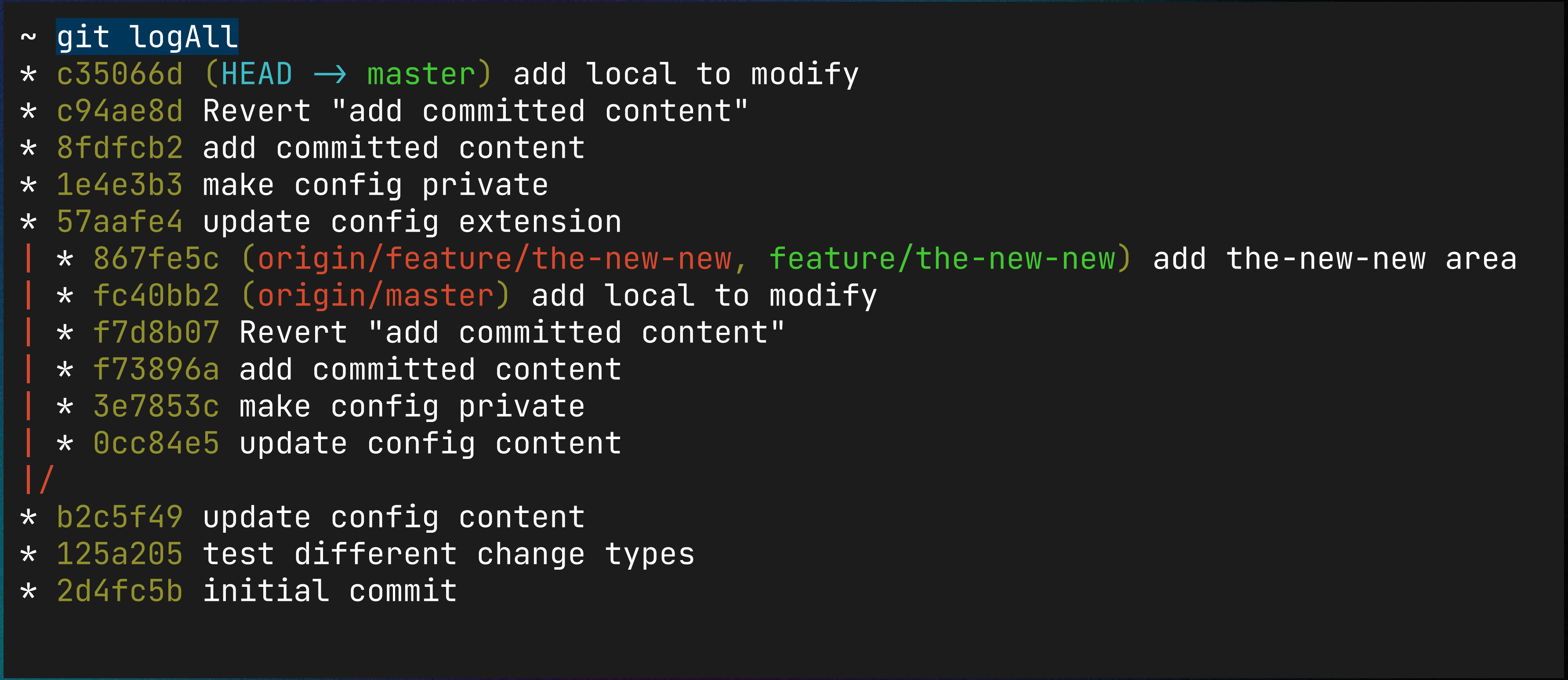
Redo commit, then `git rebase --continue`



```
~ echo "{ \"isDev\": false }" > config-prod.json
~ git add config-prod.json
~ git commit -m "update config extension"
[detached HEAD 8089209] update config extension
 2 files changed, 1 insertion(+)
 create mode 100644 config-prod.json
 rename config.txt => config.json (100%)
~ git rebase --continue
Successfully rebased and updated refs/heads/master.
```

Prep Content

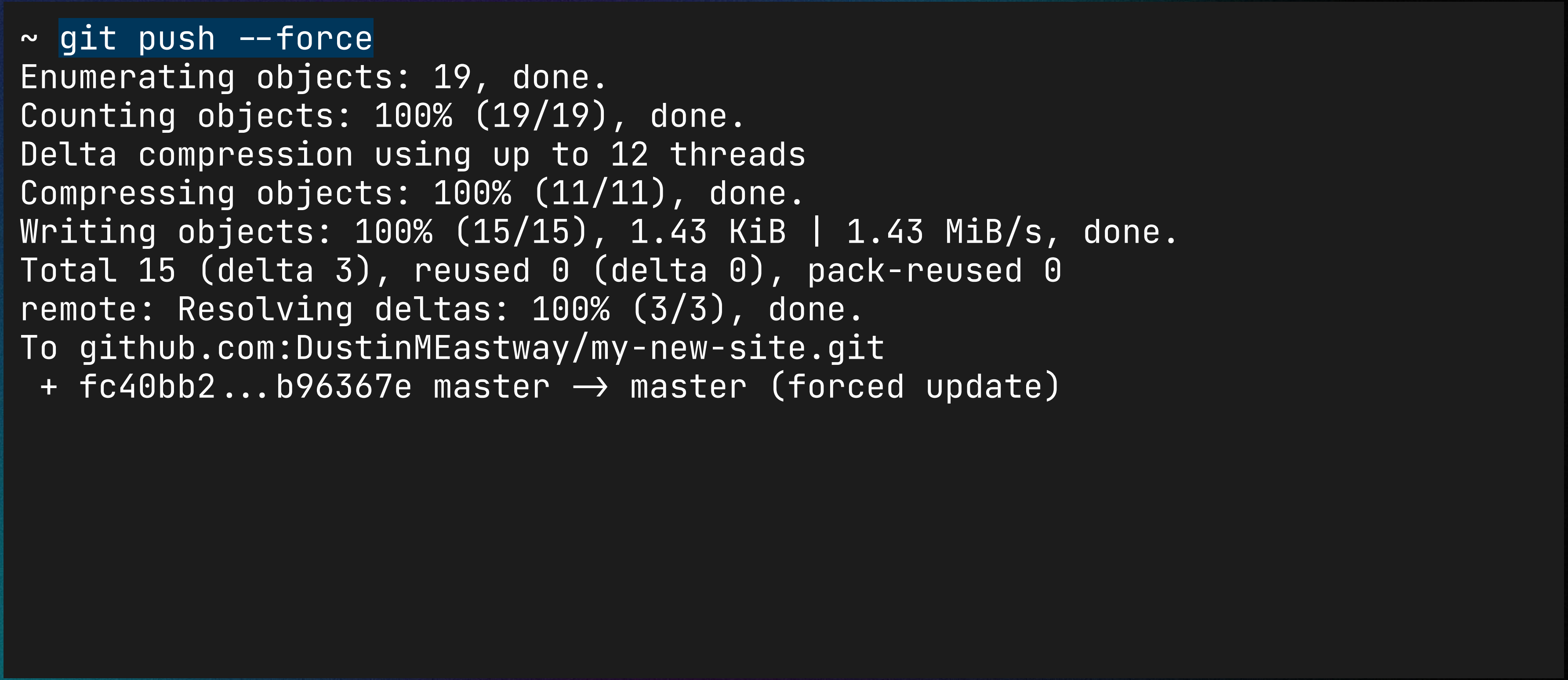
Display rebase aftermath.



```
~ git logAll
* c35066d (HEAD → master) add local to modify
* c94ae8d Revert "add committed content"
* 8fdfcb2 add committed content
* 1e4e3b3 make config private
* 57aafe4 update config extension
| * 867fe5c (origin/feature/the-new-new, feature/the-new-new) add the-new-new area
| * fc40bb2 (origin/master) add local to modify
| * f7d8b07 Revert "add committed content"
| * f73896a add committed content
| * 3e7853c make config private
| * 0cc84e5 update config content
|/
* b2c5f49 update config content
* 125a205 test different change types
* 2d4fc5b initial commit
```

Prep Content

Display rebase aftermath. Continued...

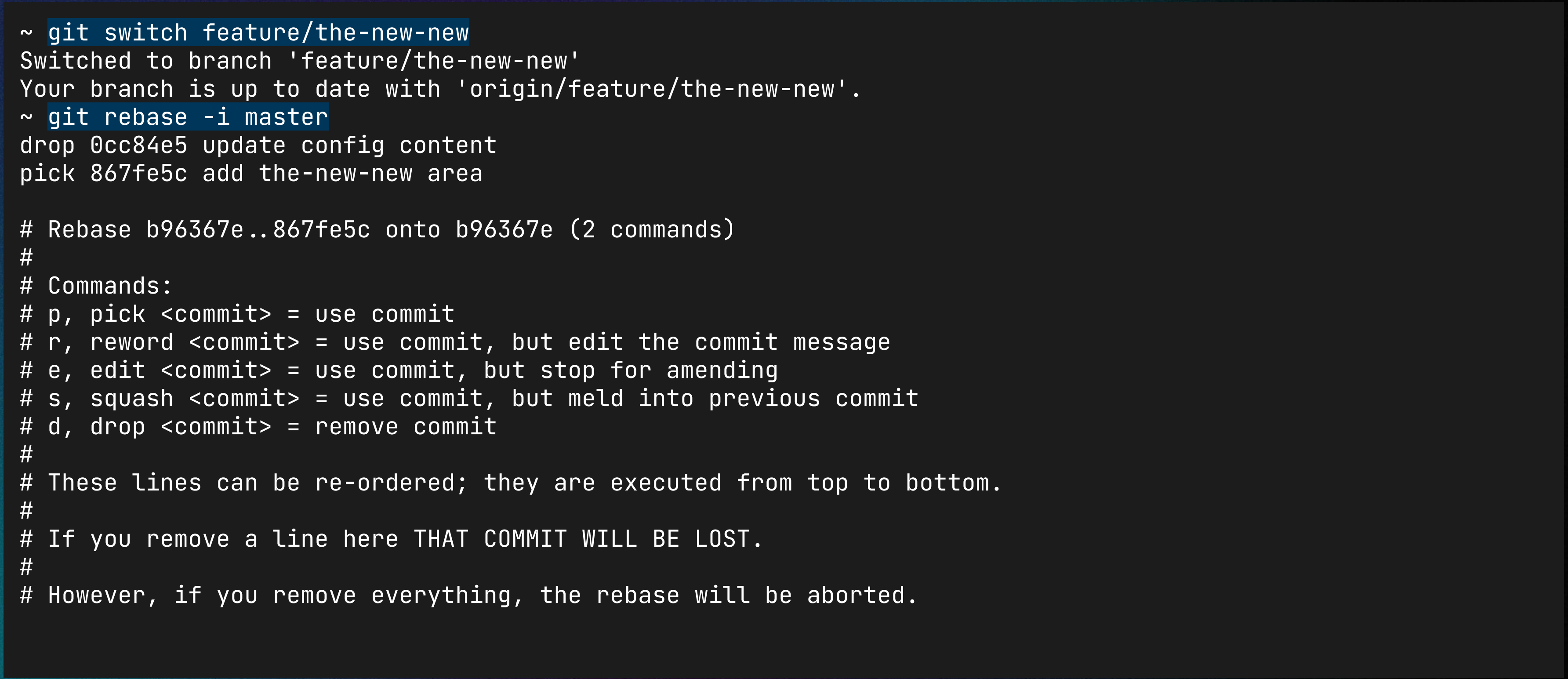


```
~ git push --force
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (15/15), 1.43 KiB | 1.43 MiB/s, done.
Total 15 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To github.com:DustinMEastway/my-new-site.git
 + fc40bb2...b96367e master -> master (forced update)
```

How Do I Combine Histories?

Interactive rebase.

git rebase -i <full branch name or commit hash>

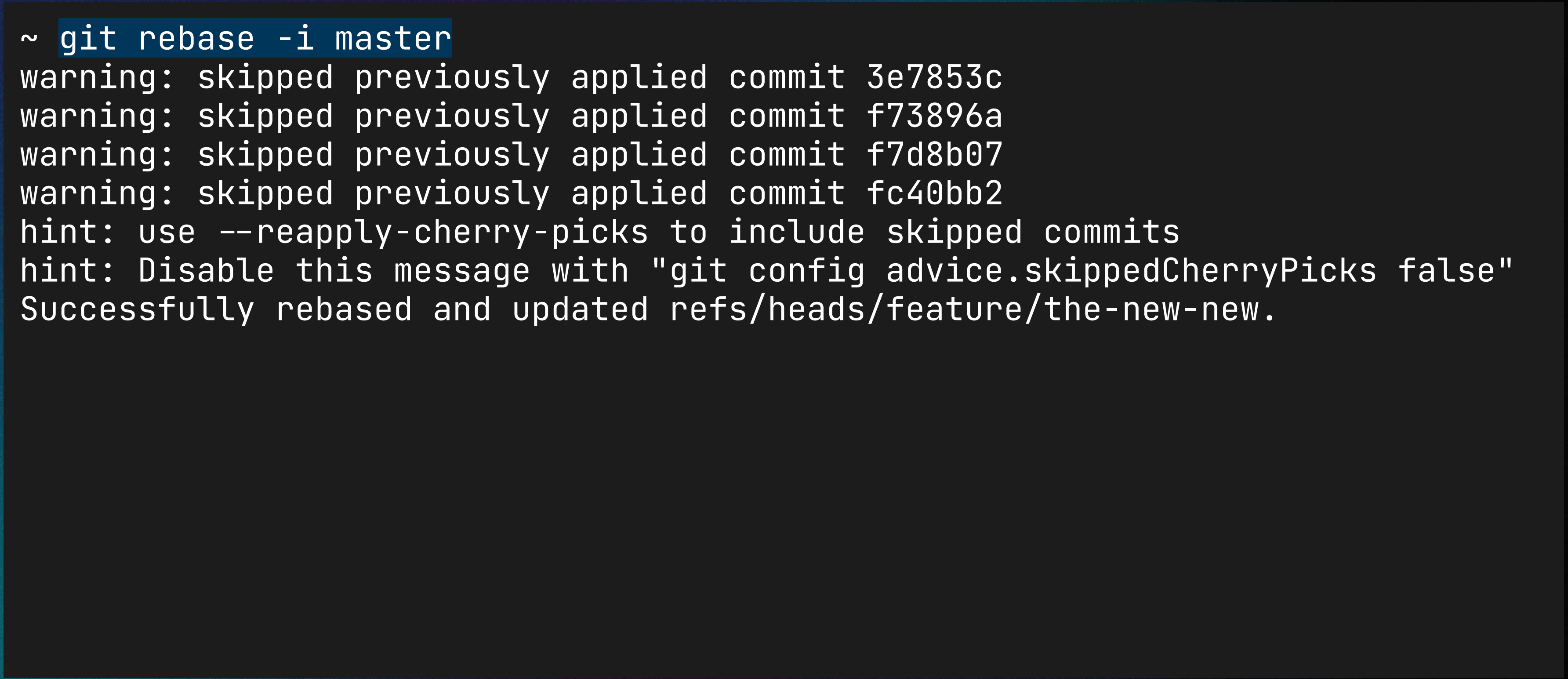


```
~ git switch feature/the-new-new
Switched to branch 'feature/the-new-new'
Your branch is up to date with 'origin/feature/the-new-new'.
~ git rebase -i master
drop 0cc84e5 update config content
pick 867fe5c add the-new-new area

# Rebase b96367e..867fe5c onto b96367e (2 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# d, drop <commit> = remove commit
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
```

How Do I Combine Histories?

Interactive rebase. Continued...

git rebase -i <full branch name or commit hash>

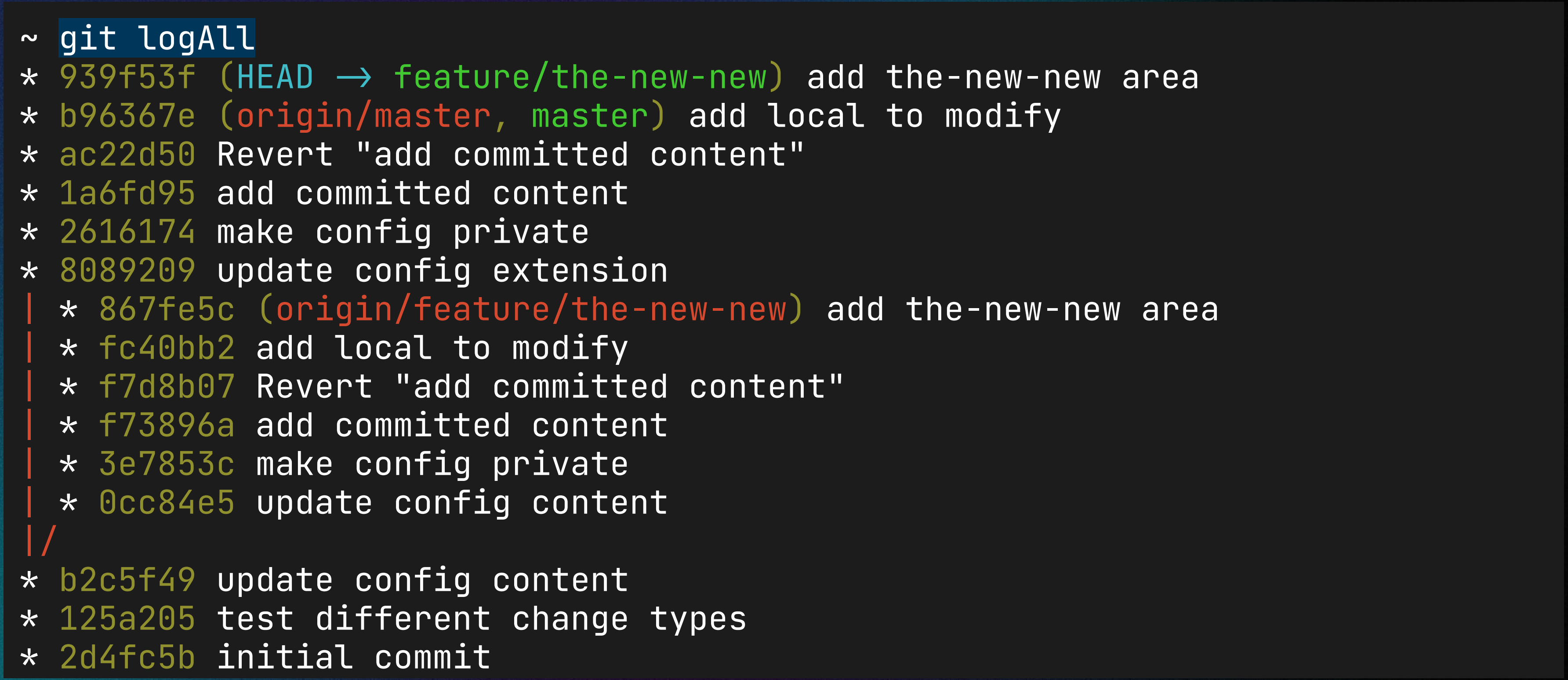


```
~ git rebase -i master
warning: skipped previously applied commit 3e7853c
warning: skipped previously applied commit f73896a
warning: skipped previously applied commit f7d8b07
warning: skipped previously applied commit fc40bb2
hint: use --reapply-cherry-picks to include skipped commits
hint: Disable this message with "git config advice.skippedCherryPicks false"
Successfully rebased and updated refs/heads/feature/the-new-new.
```

How Do I Combine Histories?

Interactive rebase. Continued...

`git rebase -i <full branch name or commit hash>`

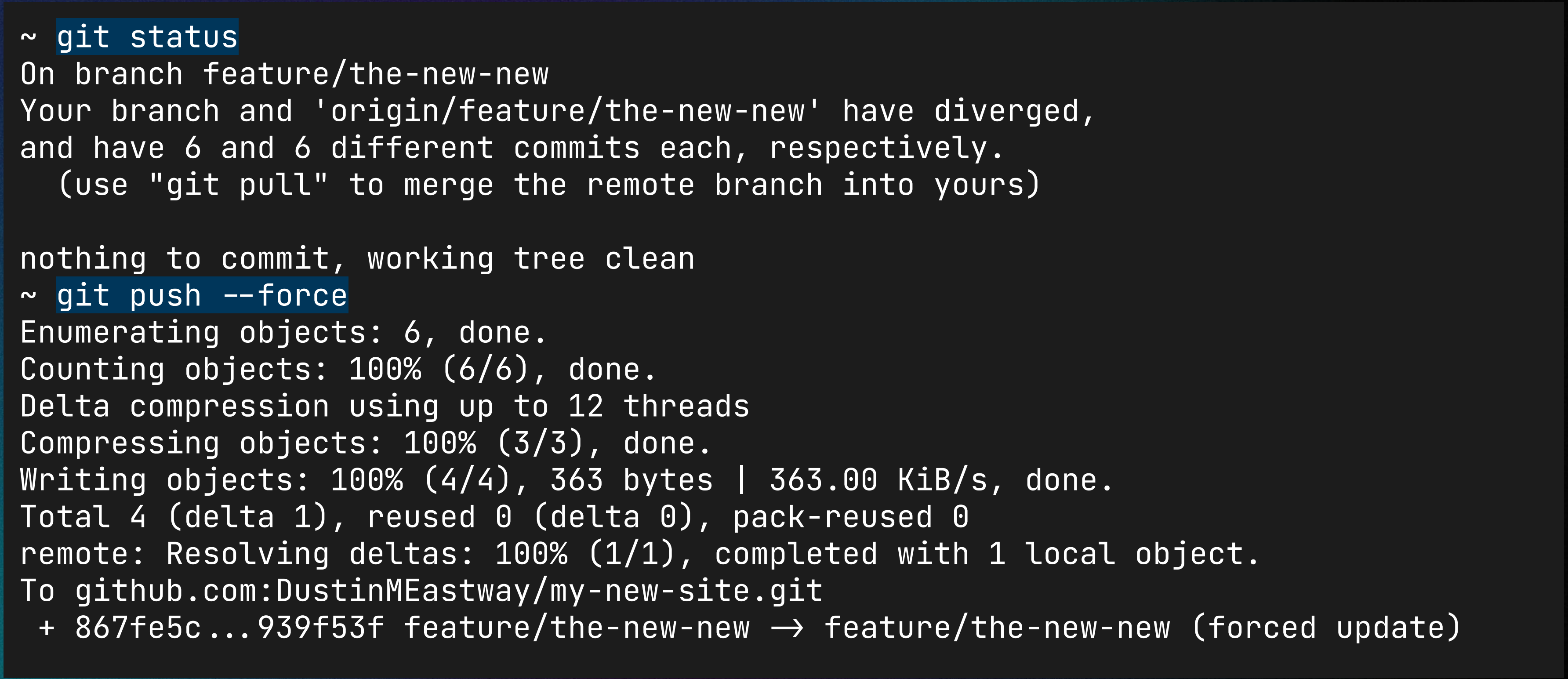


```
~ git logAll
* 939f53f (HEAD -> feature/the-new-new) add the-new-new area
* b96367e (origin/master, master) add local to modify
* ac22d50 Revert "add committed content"
* 1a6fd95 add committed content
* 2616174 make config private
* 8089209 update config extension
| * 867fe5c (origin/feature/the-new-new) add the-new-new area
| * fc40bb2 add local to modify
| * f7d8b07 Revert "add committed content"
| * f73896a add committed content
| * 3e7853c make config private
| * 0cc84e5 update config content
|/
* b2c5f49 update config content
* 125a205 test different change types
* 2d4fc5b initial commit
```

How Do I Combine Histories?

Interactive rebase. Continued...

git rebase -i <full branch name or commit hash>

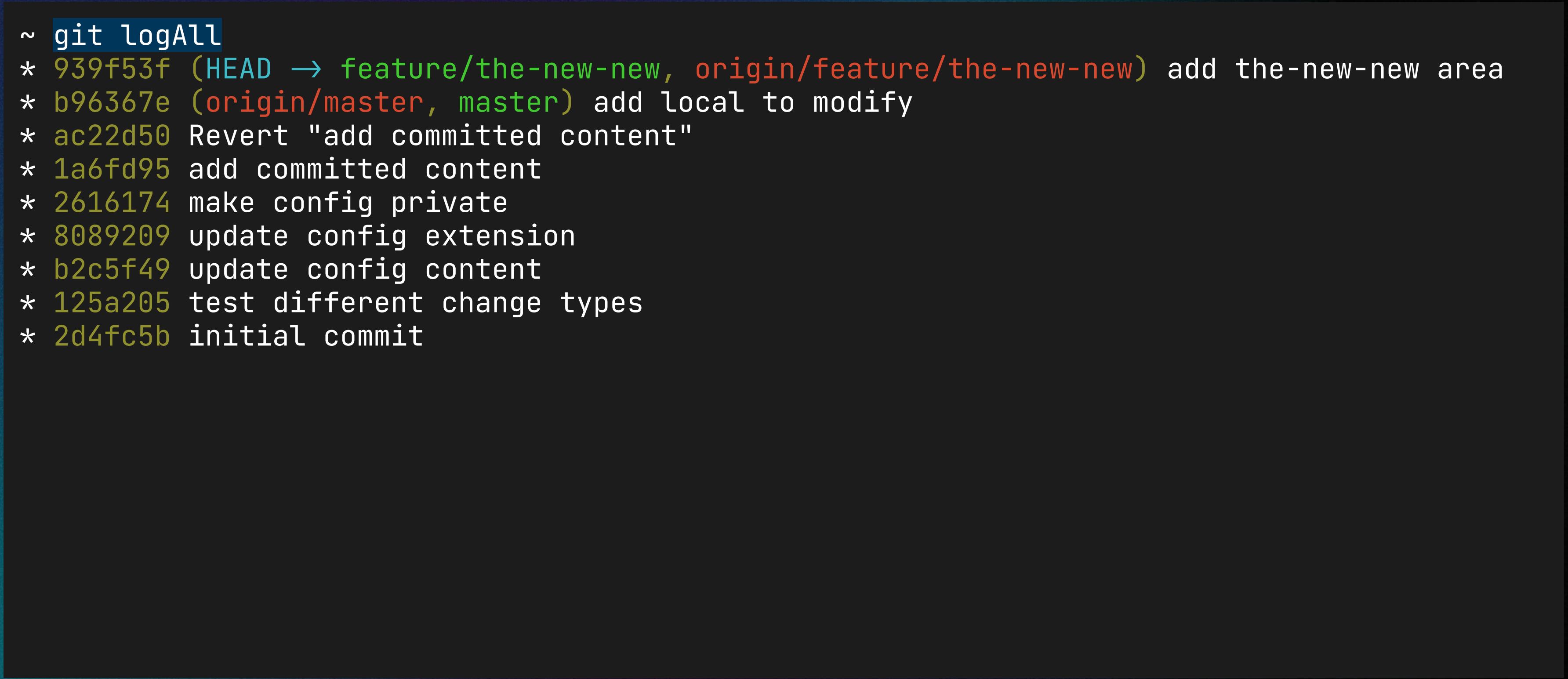


```
~ git status
On branch feature/the-new-new
Your branch and 'origin/feature/the-new-new' have diverged,
and have 6 and 6 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean
~ git push --force
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 363 bytes | 363.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:DustinMEastway/my-new-site.git
 + 867fe5c...939f53f feature/the-new-new -> feature/the-new-new (forced update)
```

How Do I Combine Histories?

Interactive rebase. Continued...

git rebase -i <full branch name or commit hash>

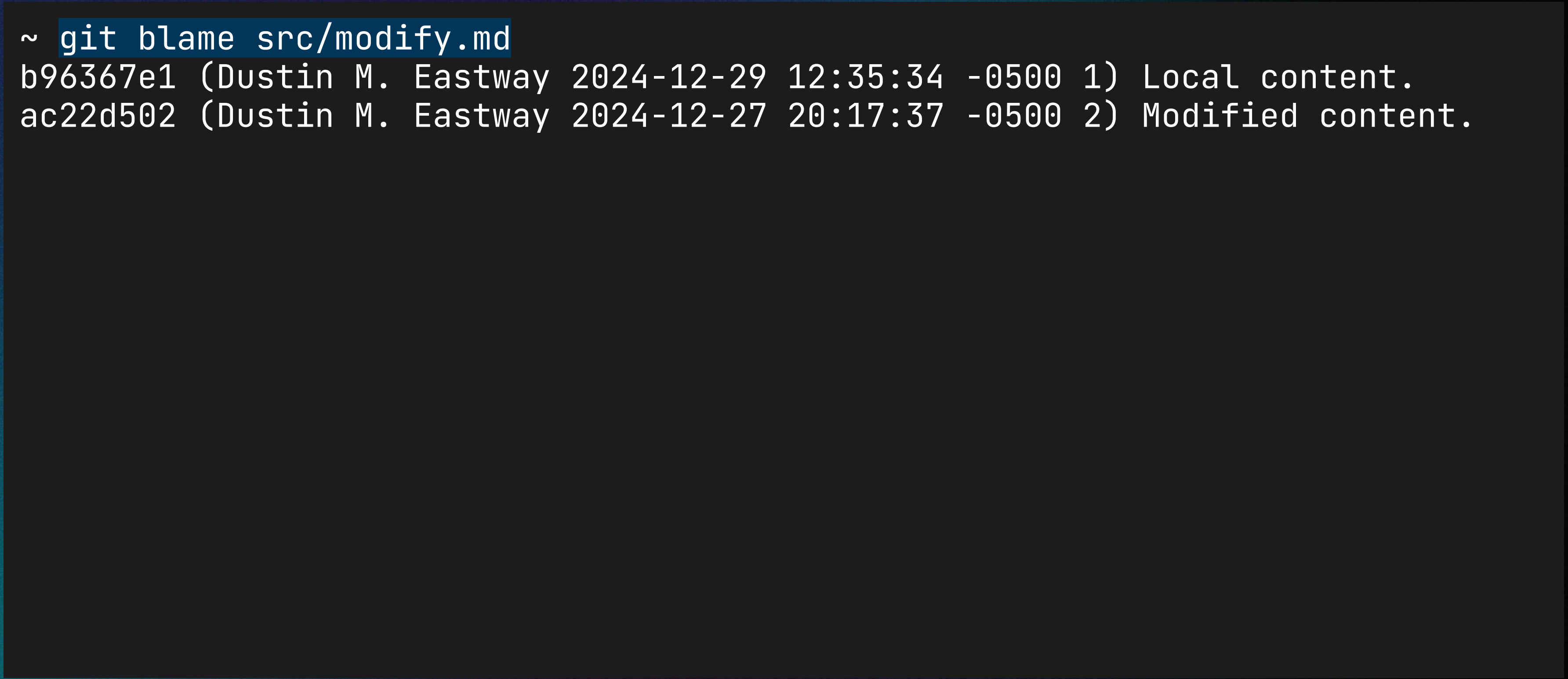


```
~ git logAll
* 939f53f (HEAD -> feature/the-new-new, origin/feature/the-new-new) add the-new-new area
* b96367e (origin/master, master) add local to modify
* ac22d50 Revert "add committed content"
* 1a6fd95 add committed content
* 2616174 make config private
* 8089209 update config extension
* b2c5f49 update config content
* 125a205 test different change types
* 2d4fc5b initial commit
```

How To View A File’s History?

History of each line.

`git blame <path to file>`

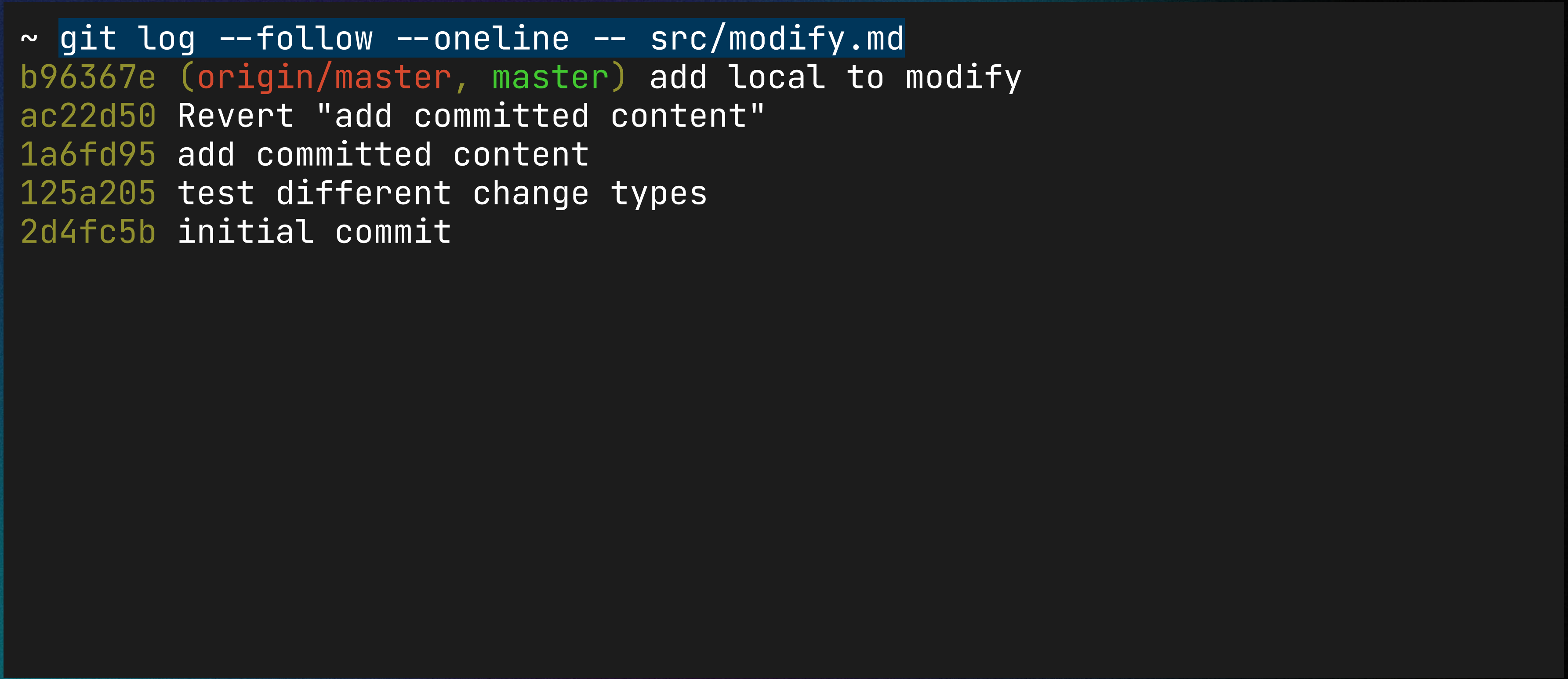


```
~ git blame src/modify.md
b96367e1 (Dustin M. Eastway 2024-12-29 12:35:34 -0500 1) Local content.
ac22d502 (Dustin M. Eastway 2024-12-27 20:17:37 -0500 2) Modified content.
```

How To View A File’s History?

History of each line.

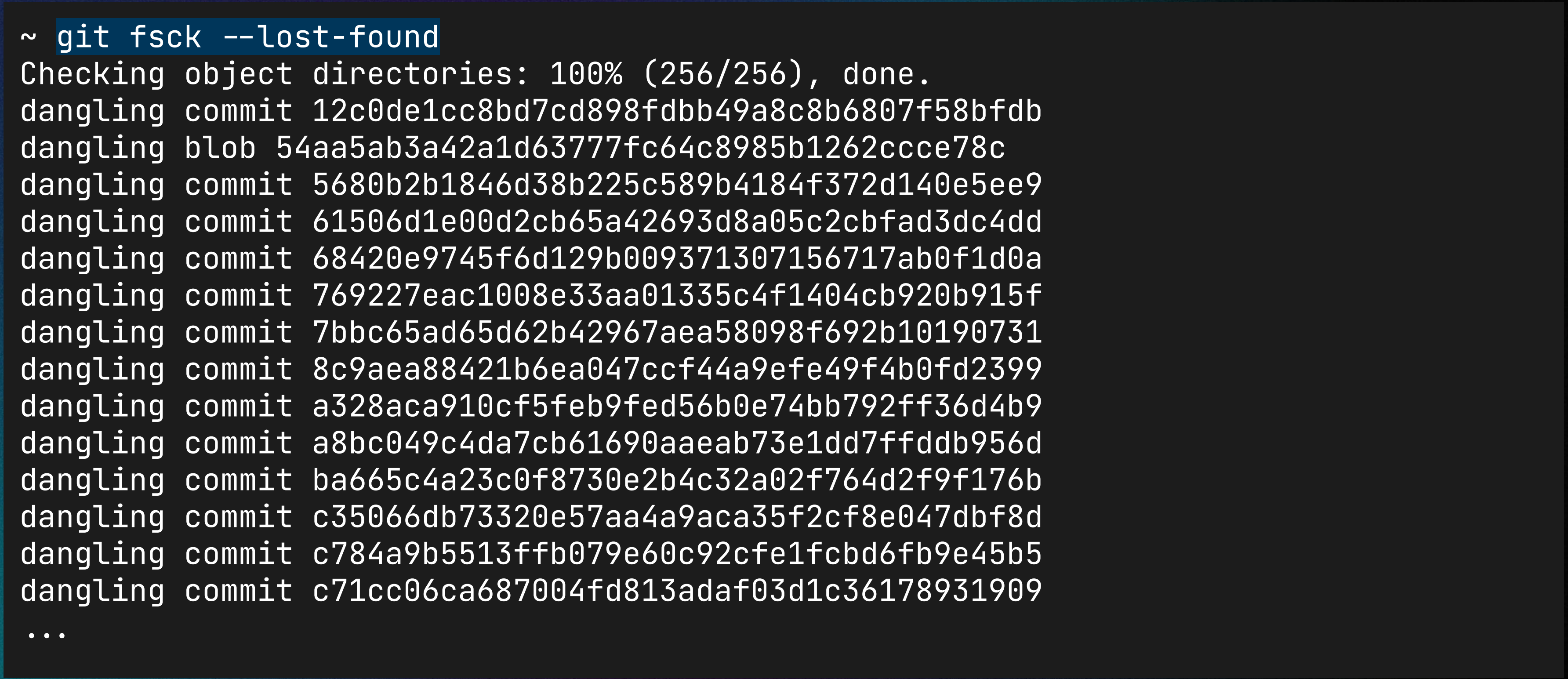
git log [--follow] [--oneline] -- <path to file>



```
~ git log --follow --oneline -- src/modify.md
b96367e (origin/master, master) add local to modify
ac22d50 Revert "add committed content"
1a6fd95 add committed content
125a205 test different change types
2d4fc5b initial commit
```

How Do I Find Branchless Commits?

`git fsck --lost-found`



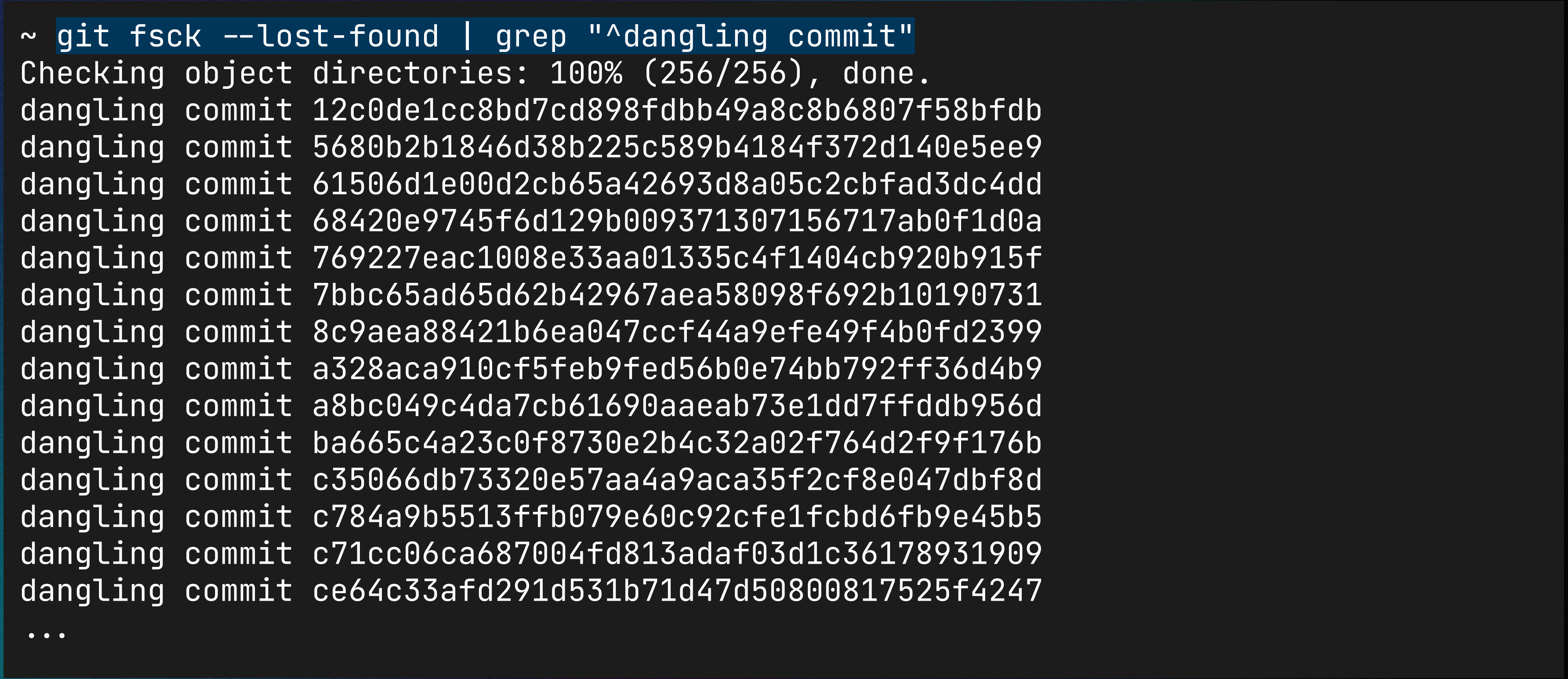
```
~ git fsck --lost-found
Checking object directories: 100% (256/256), done.
dangling commit 12c0de1cc8bd7cd898fdbb49a8c8b6807f58bfdb
dangling blob 54aa5ab3a42a1d63777fc64c8985b1262ccce78c
dangling commit 5680b2b1846d38b225c589b4184f372d140e5ee9
dangling commit 61506d1e00d2cb65a42693d8a05c2cbfad3dc4dd
dangling commit 68420e9745f6d129b009371307156717ab0f1d0a
dangling commit 769227eac1008e33aa01335c4f1404cb920b915f
dangling commit 7bbc65ad65d62b42967aea58098f692b10190731
dangling commit 8c9aea88421b6ea047ccf44a9efe49f4b0fd2399
dangling commit a328aca910cf5feb9fed56b0e74bb792ff36d4b9
dangling commit a8bc049c4da7cb61690aaeab73e1dd7ffddb956d
dangling commit ba665c4a23c0f8730e2b4c32a02f764d2f9f176b
dangling commit c35066db73320e57aa4a9aca35f2cf8e047dbf8d
dangling commit c784a9b5513ffb079e60c92cfe1fcbd6fb9e45b5
dangling commit c71cc06ca687004fd813adaf03d1c36178931909
...
```

Source: Luke Scott's site

How Do I Find Branchless Commits?

Continued...

`git fsck --lost-found`



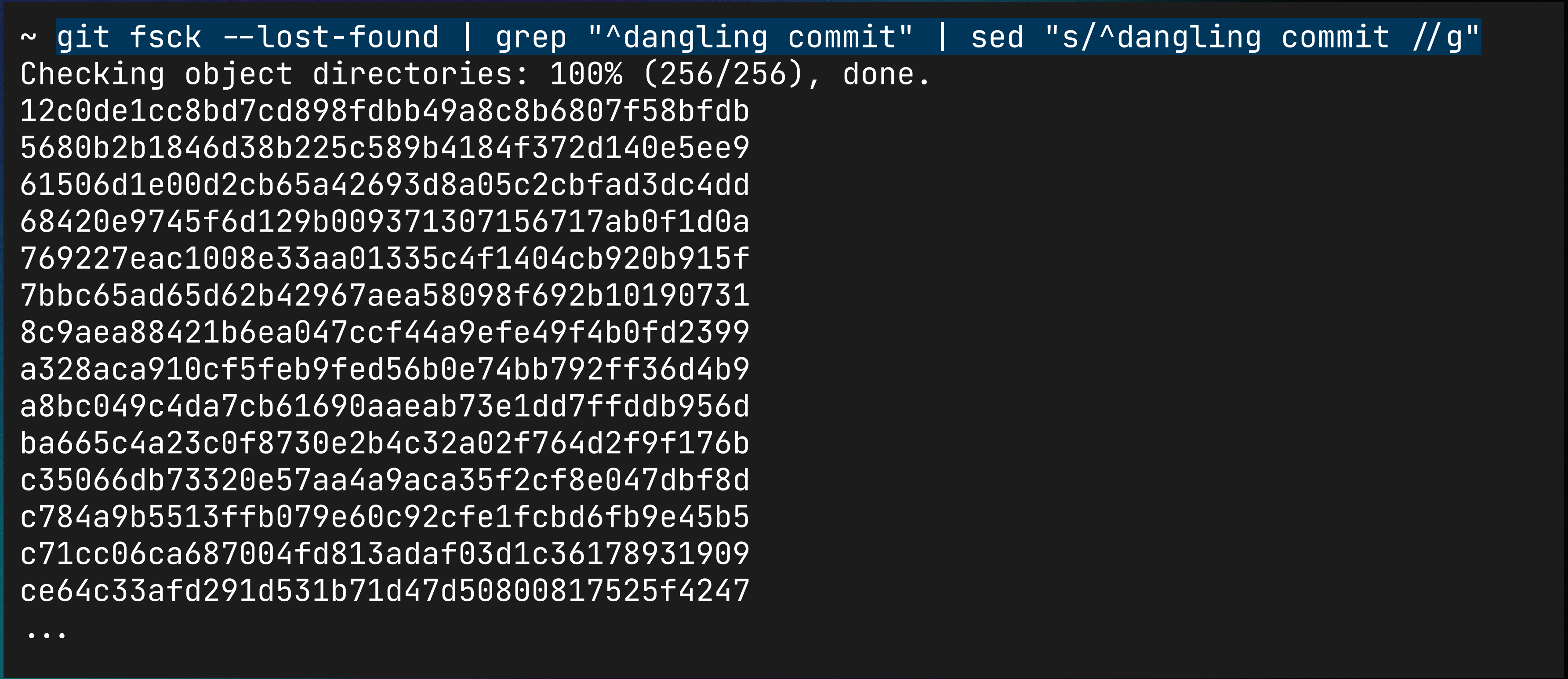
```
~ git fsck --lost-found | grep "^dangling commit"
Checking object directories: 100% (256/256), done.
dangling commit 12c0de1cc8bd7cd898fdbb49a8c8b6807f58bfdb
dangling commit 5680b2b1846d38b225c589b4184f372d140e5ee9
dangling commit 61506d1e00d2cb65a42693d8a05c2cbfad3dc4dd
dangling commit 68420e9745f6d129b009371307156717ab0f1d0a
dangling commit 769227eac1008e33aa01335c4f1404cb920b915f
dangling commit 7bbc65ad65d62b42967aea58098f692b10190731
dangling commit 8c9aea88421b6ea047ccf44a9efe49f4b0fd2399
dangling commit a328aca910cf5feb9fed56b0e74bb792ff36d4b9
dangling commit a8bc049c4da7cb61690aaeab73e1dd7ffddb956d
dangling commit ba665c4a23c0f8730e2b4c32a02f764d2f9f176b
dangling commit c35066db73320e57aa4a9aca35f2cf8e047dbf8d
dangling commit c784a9b5513ffb079e60c92cfe1fcbd6fb9e45b5
dangling commit c71cc06ca687004fd813adaf03d1c36178931909
dangling commit ce64c33afd291d531b71d47d50800817525f4247
...
```

Source: Luke Scott's site

How Do I Find Branchless Commits?

Continued...

`git fsck --lost-found`



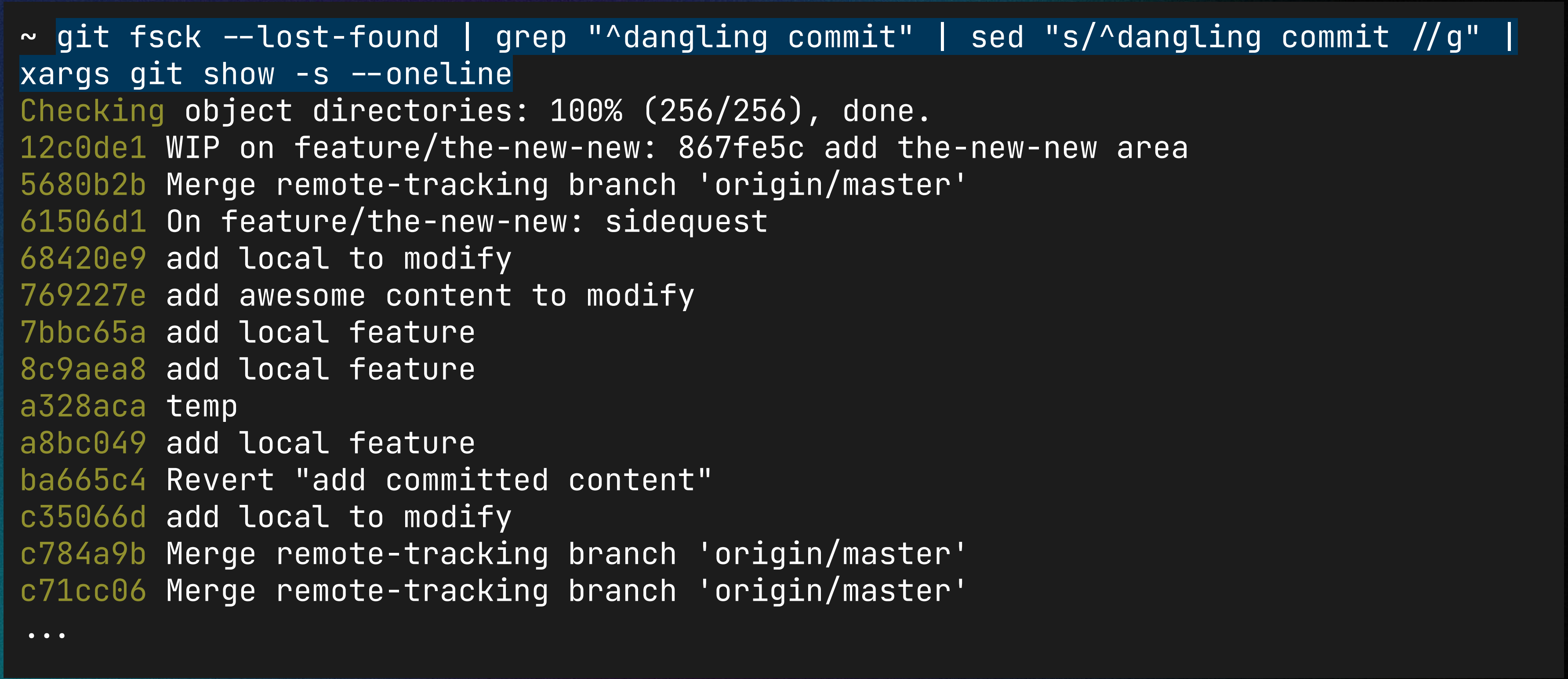
```
~ git fsck --lost-found | grep "^dangling commit" | sed "s/^dangling commit //g"
Checking object directories: 100% (256/256), done.
12c0de1cc8bd7cd898fdbb49a8c8b6807f58bfdb
5680b2b1846d38b225c589b4184f372d140e5ee9
61506d1e00d2cb65a42693d8a05c2cbfad3dc4dd
68420e9745f6d129b00937130715671 7ab0f1d0a
769227eac1008e33aa01335c4f1404cb920b915f
7bbc65ad65d62b42967aea58098f692b10190731
8c9aea88421b6ea047ccf44a9efe49f4b0fd2399
a328aca910cf5feb9fed56b0e74bb792ff36d4b9
a8bc049c4da7cb61690aaeab73e1dd7ffddb956d
ba665c4a23c0f8730e2b4c32a02f764d2f9f176b
c35066db73320e57aa4a9aca35f2cf8e047dbf8d
c784a9b5513ffb079e60c92cfe1fcbd6fb9e45b5
c71cc06ca687004fd813adaf03d1c36178931909
ce64c33afd291d531b71d47d50800817525f4247
...
```

Source: Luke Scott's site

How Do I Find Branchless Commits?

Continued...

`git fsck --lost-found`



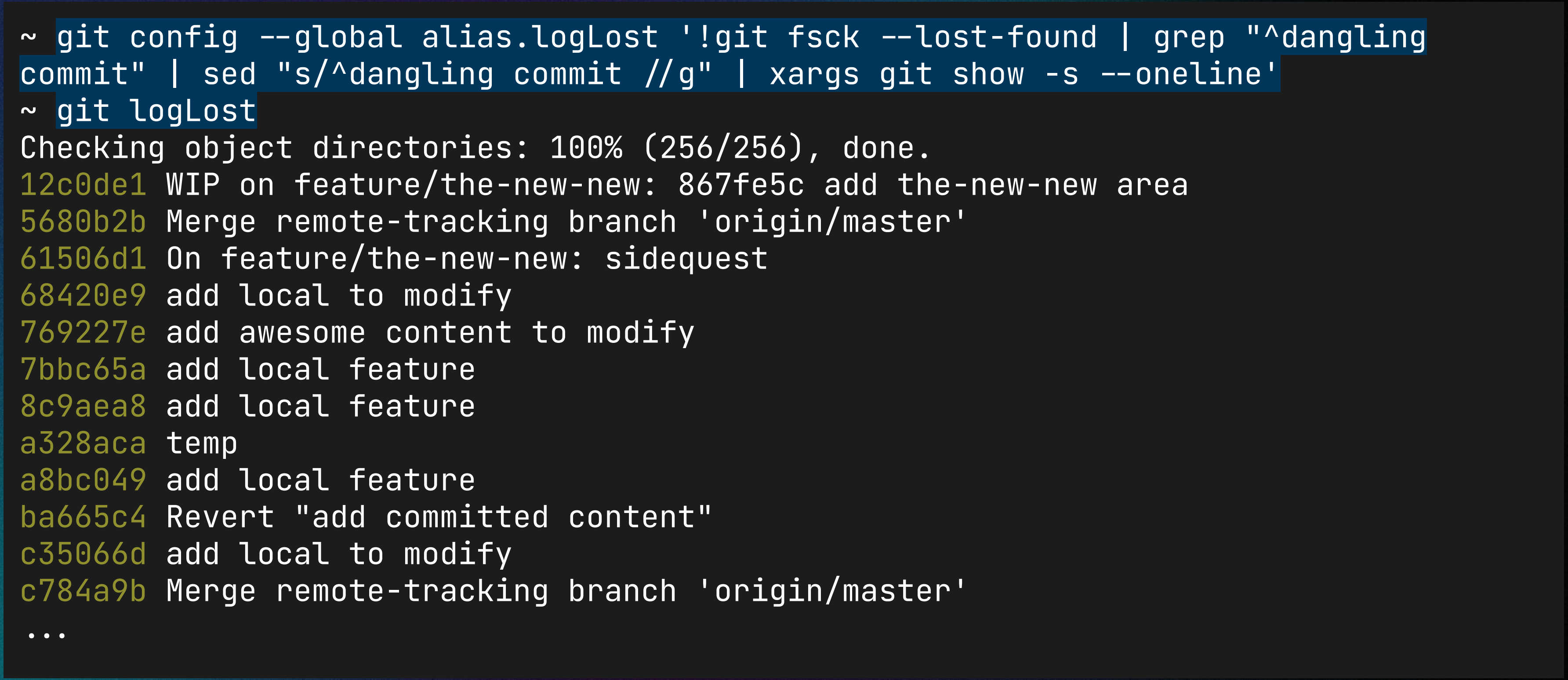
```
~ git fsck --lost-found | grep "^dangling commit" | sed "s/^dangling commit //g" |
xargs git show -s --oneline
Checking object directories: 100% (256/256), done.
12c0de1 WIP on feature/the-new-new: 867fe5c add the-new-new area
5680b2b Merge remote-tracking branch 'origin/master'
61506d1 On feature/the-new-new: sidequest
68420e9 add local to modify
769227e add awesome content to modify
7bbc65a add local feature
8c9aea8 add local feature
a328aca temp
a8bc049 add local feature
ba665c4 Revert "add committed content"
c35066d add local to modify
c784a9b Merge remote-tracking branch 'origin/master'
c71cc06 Merge remote-tracking branch 'origin/master'
...
```

Source: Luke Scott's site

How Do I Find Branchless Commits?

Continued...

`git fsck --lost-found`



```
~ git config --global alias.logLost '!git fsck --lost-found | grep "^dangling
commit" | sed "s/^dangling commit //g" | xargs git show -s --oneline'
~ git logLost
Checking object directories: 100% (256/256), done.
12c0de1 WIP on feature/the-new-new: 867fe5c add the-new-new area
5680b2b Merge remote-tracking branch 'origin/master'
61506d1 On feature/the-new-new: sidequest
68420e9 add local to modify
769227e add awesome content to modify
7bbc65a add local feature
8c9aea8 add local feature
a328aca temp
a8bc049 add local feature
ba665c4 Revert "add committed content"
c35066d add local to modify
c784a9b Merge remote-tracking branch 'origin/master'
...
```

Merge/Pull Request Tips

Questions?