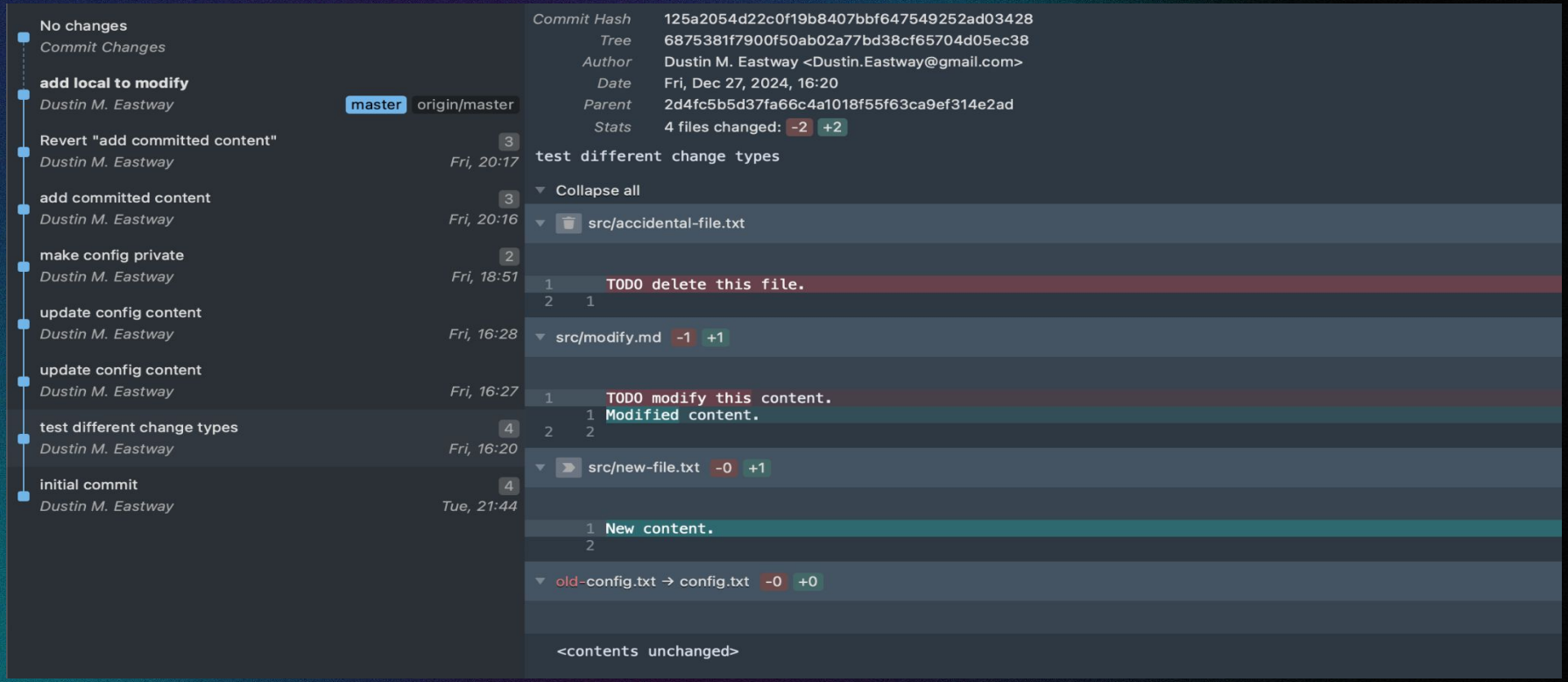
# Git On My Level

## Bridging Knowledge with Practical Solutions

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

Dustin M. Eastway
CodeMash
2025-01-17

# Term: 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
TODO delete this file.
src/modify.md -1 +1
TODO modify this content.
Modified content.
src/new-file.txt -0 +1
New content.
old-config.txt → config.txt -0 +0
<contents unchanged>

# 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
```

# 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)>]
```

# How Do I Display Commits?

# 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?

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
```

# 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
```

# 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
```

# 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 Display Commits?

## Questions?

# How Do I Configure 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 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 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 Configure Git?

## Questions?

# How Do I Find Branchless Commits?

# 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" | cut -c 17-
Checking object directories: 100% (256/256), done.
12c0de1cc8bd7cd898fdbb49a8c8b6807f58bfdb
5680b2b1846d38b225c589b4184f372d140e5ee9
61506d1e00d2cb65a42693d8a05c2cbfad3dc4dd
68420e9745f6d129b009371307156717ab0f1d0a
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" | cut -c 17- | 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" | cut -c 17- | 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'
...
```

# How Do I Find Branchless Commits?

## Questions?

# Warning: Dangers Of Moving Files

# Warning: Dangers Of Moving Files

The `git mv` & `git rm` commands are a lie!!!

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

nothing to commit, working tree clean
~ git mv config-prod.json config.json
~ cat src/new-file.txt > config.json
~ git add config.json
~ git rm src/new-file.txt
rm 'src/new-file.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)
  deleted:    config-prod.json
  renamed:    src/new-file.txt → config.json
```

# Warning: Dangers Of Moving Files

## Questions?

# How To View A File’s History?

# 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 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?

## Questions?

# How Do I Display Changes?

# 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 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.
```

# 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 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?

## Questions?

# How Do I Ignore Files?

**Modified, staged, & committed file(s).**

# How Do I Combine Histories?

# 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"
```

# 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"
```

# 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 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 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. 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?

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 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?

## Questions?

# How Do I “Modify A Commit”?

# Prep Content

Display incorrect commit.

```
~ 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
```

# 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
```

# How Do I “Modify A Commit”?

## Questions?

# Merge/Pull Request Tips

# Questions?