

Git Internals

A brownbag workshop at



by Seth House

@whiteinge
seth@eseth.com

Hands-on; Ask Questions!

Hands-on; Ask Questions!

- `brew install graphviz`
- `apt-get install graphviz`

Hands-on; Ask Questions!

- `brew install graphviz`
- `apt-get install graphviz`
 - Windows users may want something like `wsl-open`.

Hands-on; Ask Questions!

- `brew install graphviz`
- `apt-get install graphviz`
 - Windows users may want something like `wsl-open`.
- `curl -L https://git.io/fjfQB -o /usr/local/bin/git-graph-dag`

(<https://github.com/whiteinge/dotfiles/blob/master/bin/git-graph-dag>)

Hands-on; Ask Questions!

- `brew install graphviz`
- `apt-get install graphviz`
 - Windows users may want something like `wsl-open`.
- `curl -L https://git.io/fjfQB -o /usr/local/bin/git-graph-dag`
(<https://github.com/whiteinge/dotfiles/blob/master/bin/git-graph-dag>)
- Wraps `git rev-list <commit>...`.

Hands-on; Ask Questions!

- `brew install graphviz`
- `apt-get install graphviz`
 - Windows users may want something like `wsl-open`.
- `curl -L https://git.io/fjfQB -o /usr/local/bin/git-graph-dag`
(<https://github.com/whiteinge/dotfiles/blob/master/bin/git-graph-dag>)
- Wraps `git rev-list <commit>...`.
- Run `git graph-dag` without args to see usage.

Can also be called directly. (`chmod +x git-graph-dat`)

Hands-on; Ask Questions!

- `brew install graphviz`
- `apt-get install graphviz`
 - Windows users may want something like `wsl-open`.
- `curl -L https://git.io/fjfQB -o /usr/local/bin/git-graph-dag`

(<https://github.com/whiteinge/dotfiles/blob/master/bin/git-graph-dag>)

- Wraps `git rev-list <commit> ...`.
- Run `git graph-dag` without args to see usage.

Can also be called directly. (`chmod +x git-graph-dat`)

- Note, possible (likely, even) to generate multi-gigabyte images.
 - `ctrl-c` to abort.
 - Limit commit ranges.

Example

```
git graph-dag master~10..master origin/master~10..origin/master \  
  | dot -Tpng \  
  | open -a Preview -f
```

Graphs & Pointers

Branches

Branches

- `find .git/refs`

Branches

- `find .git/refs`
- `cat .git/refs/heads/master`

Branches

- `find .git/refs`
- `cat .git/refs/heads/master`
- `less .git/packed-refs`

Refs

- Branches (local and remote).
- Tags.
- HEAD.

show

show

- `git show @{u}..`

show

- `git show @{u}..`
- `show --summary --stat --pretty=fuller --patch`

show

- `git show @{u}..`
- `show --summary --stat --pretty=fuller --patch`
- `printf 'foo\n' | git hash-object -w`

Local and remote refs

- `git fetch`

Local and remote refs

- `git fetch`
- `git branch --set-upstream-to=origin/master`

Rebasing

Example

```
git init testrepo
cd testrepo
for commit in A B; do git stub $commit; done
git checkout -b feature HEAD~1
for commit in C D E; do git stub $commit; done

# git graph-dag master feature
# git rebase master
# git graph-dag master feature
# git graph-dag master feature <old-E-ref>
```

Interactive

```
git init testrepo  
cd testrepo  
for commit in A B C D E F G; do git stub $commit; done  
git rebase -i HEAD~5
```


Reflog

- `git --no-pager reflog --date=relative`
- The reflog counts as a reference!
- Persists for 90-days (default).

Merging

Merge Conflicts

```
<<<<<< HEAD
twas brillig, and the slithy toves
Did gyre and gimble in the wabe:
all mimsy were the borogoves,
And the mome raths outgrabe.
=====
'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe:
All mimsy were the borogroves
And the mome raths outgabe.
>>>>>> branchA
```

Call to Action:
Experiment!