Reflogs

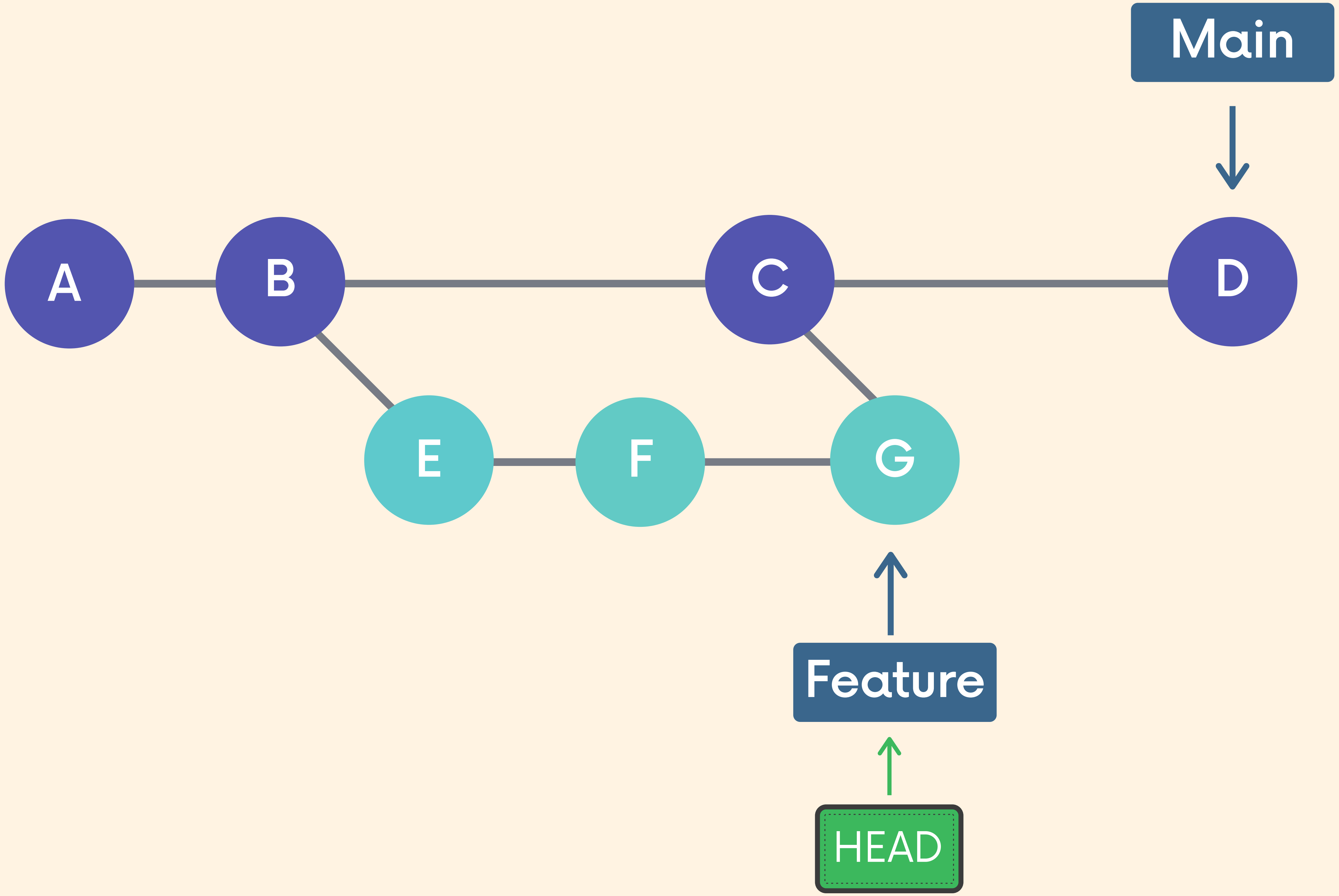
reflogs

Git keeps a record of when the tips of branches and other references were updated in the repo.

We can view and update these reference logs using the **git reflog** command

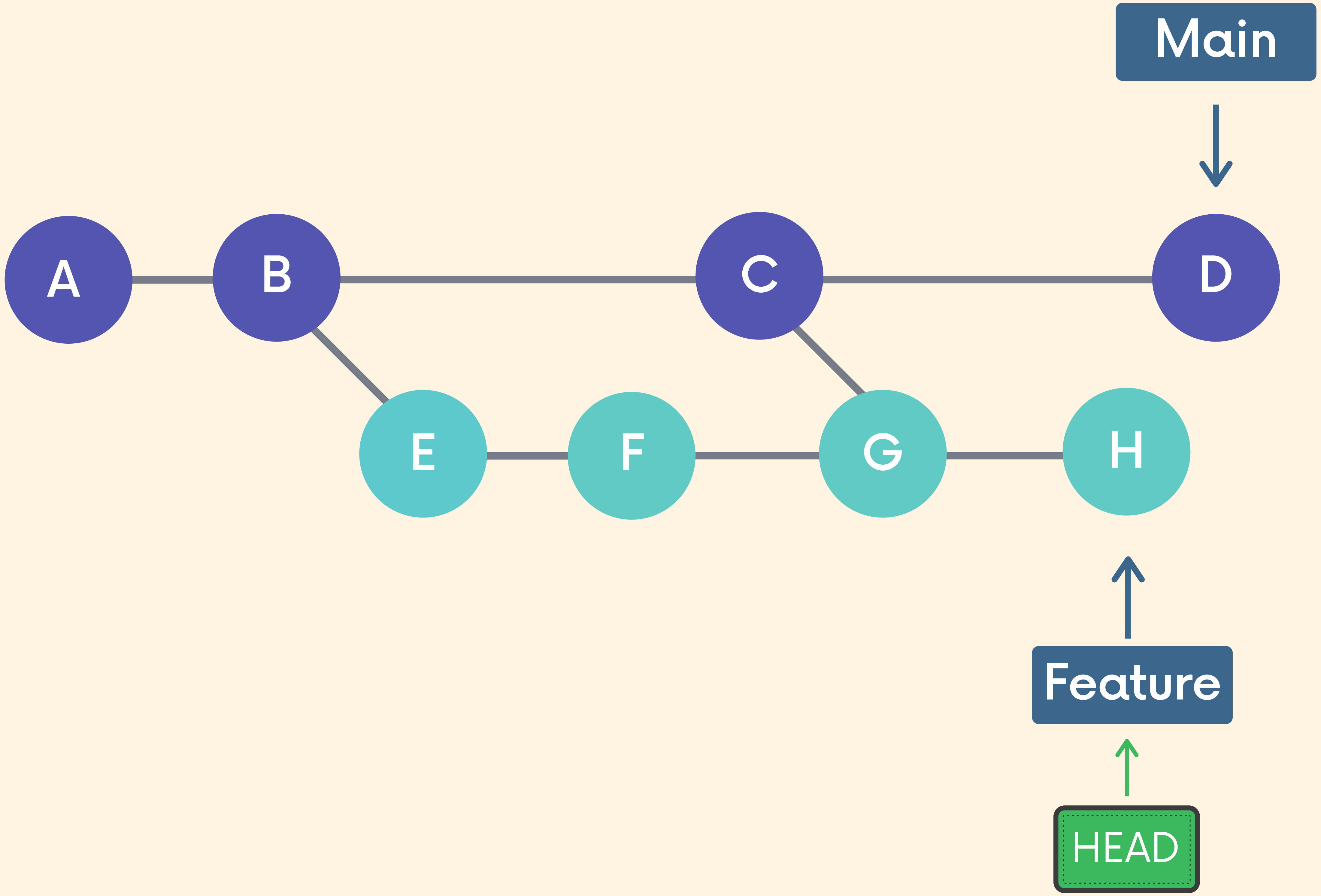
HEAD Reflog

- Commit G - New commit



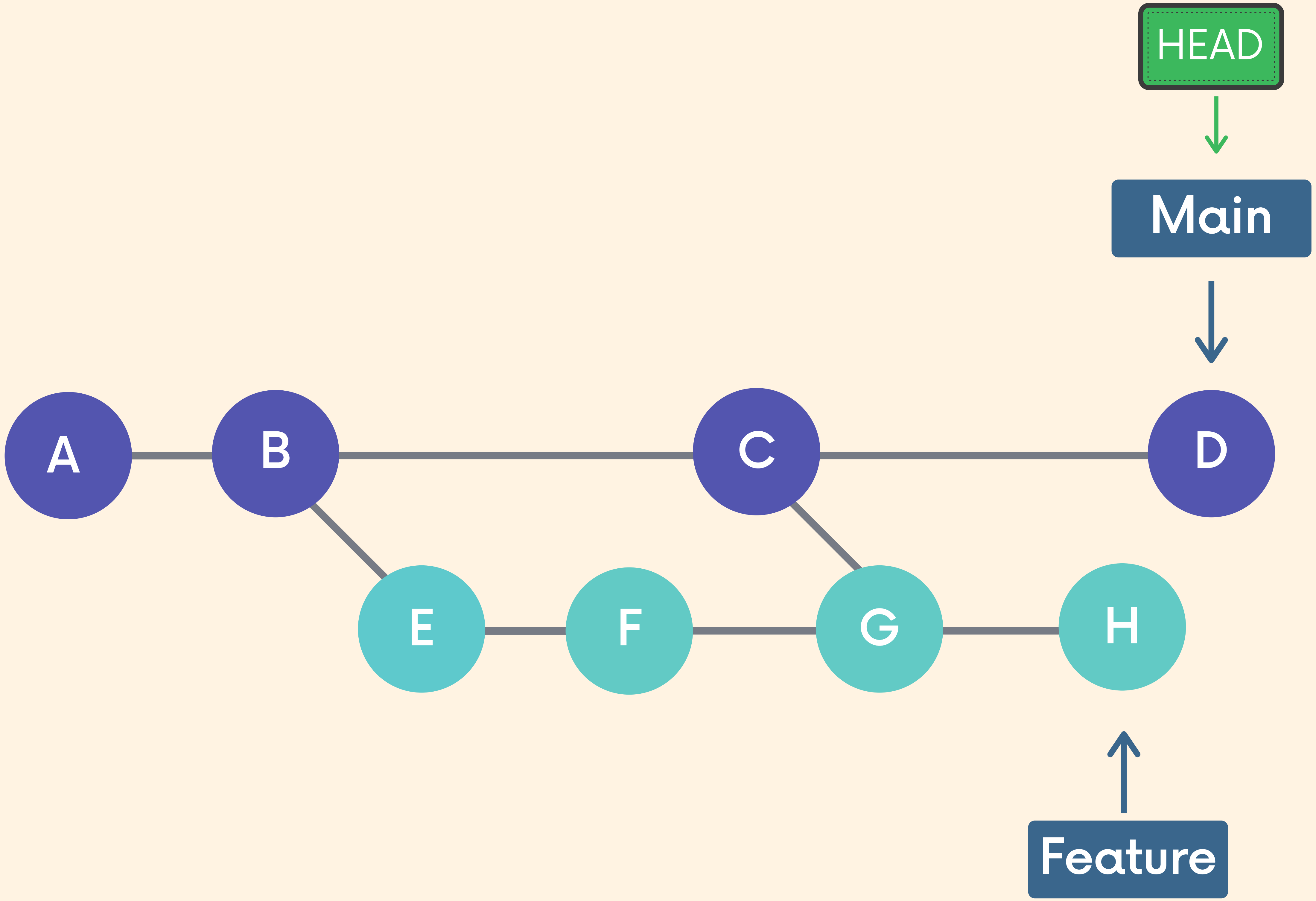
HEAD Reflog

- Commit G - New commit
- Commit H - New commit



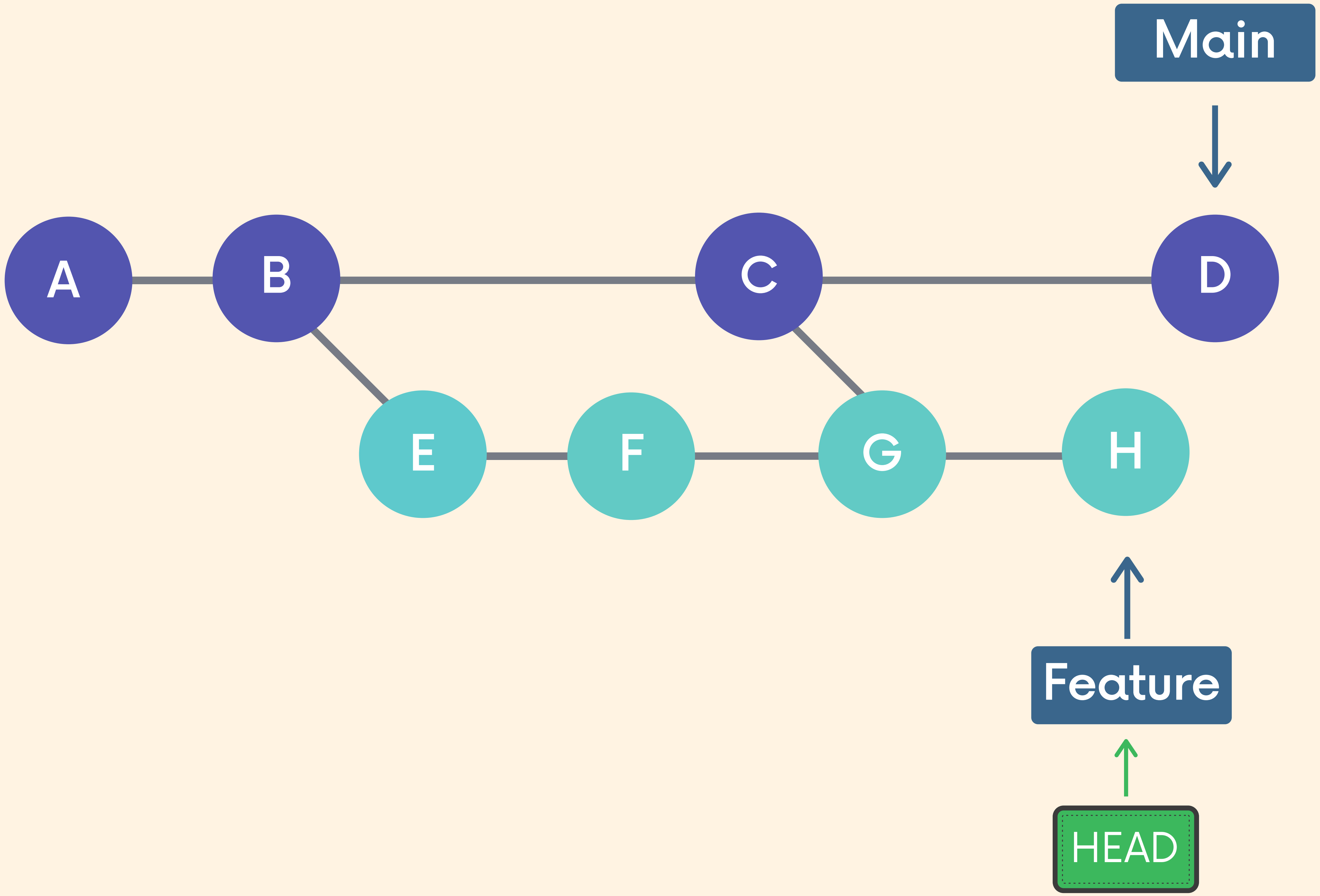
HEAD Reflog

- Commit G - New commit
- Commit H - New commit
- Commit D - Switched to main from feature



HEAD Reflog

- Commit G - New commit
- Commit H - New commit
- Commit D - Switched to main from feature
- Commit H - Switched to feature from main



Limitations!

Git only keeps reflogs on your **local** activity. They are not shared with collaborators.

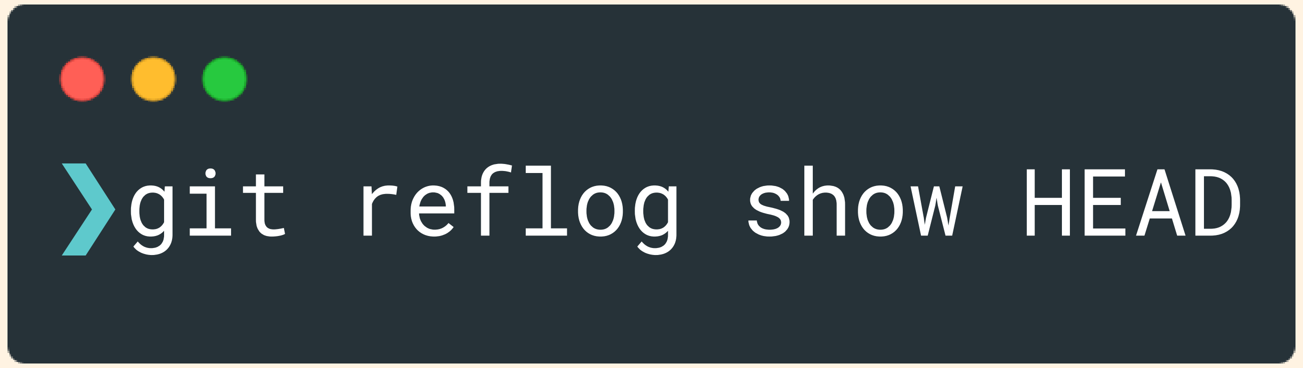
Reflogs also expire. Git cleans out old entries after around 90 days, though this can be configured.

Git Reflog

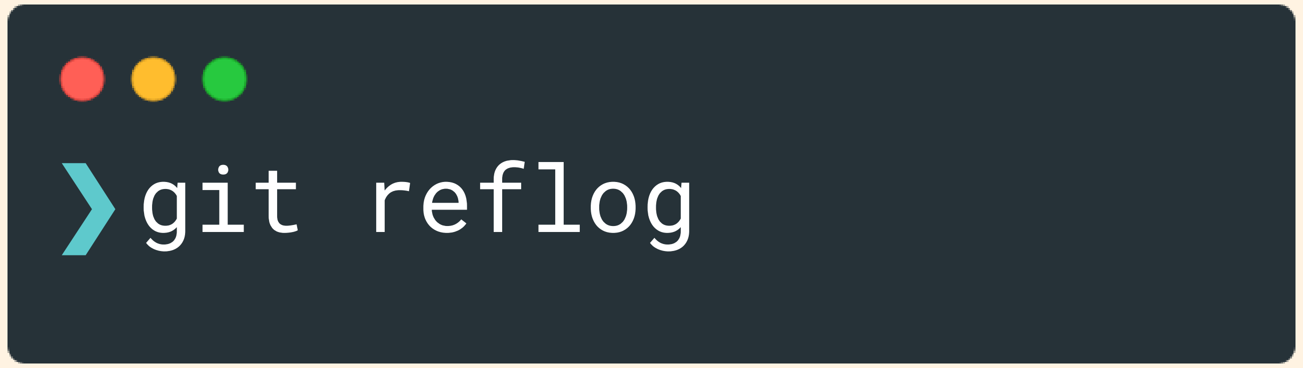
The git reflog command accepts subcommands **show**, **expire, delete**, and **exists.** Show is the only commonly used variant, and it is the default subcommand.

git reflog show will show the log of a specific reference (it defaults to HEAD).

For example, to view the logs for the tip of the main branch we could run **git reflog show main.**



```
❯git reflog show HEAD
```



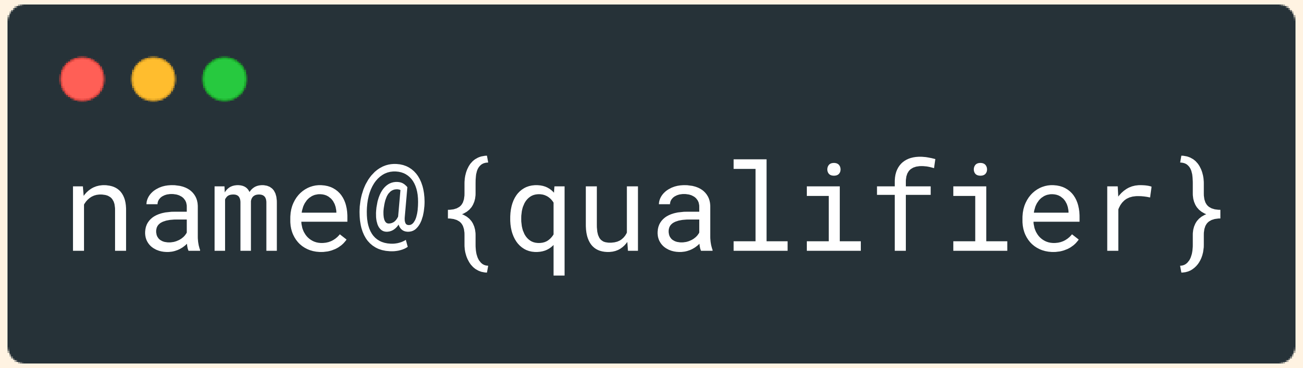
```
❯git reflog
```

HEAD Reflog

```
HEAD@{0}: commit: fix some bug
HEAD@{1}: checkout: moving from main to bugfix
HEAD@{2}: commit: another commit yay
HEAD@{3}: commit: add some feature
...
HEAD@{21}: clone: from
https://github.com/facebook/react.git
```

Reflog References

We can access specific git refs is **name@{qualifier}**. We can use this syntax to access specific ref pointers and can pass them to other commands including checkout, reset, and merge.

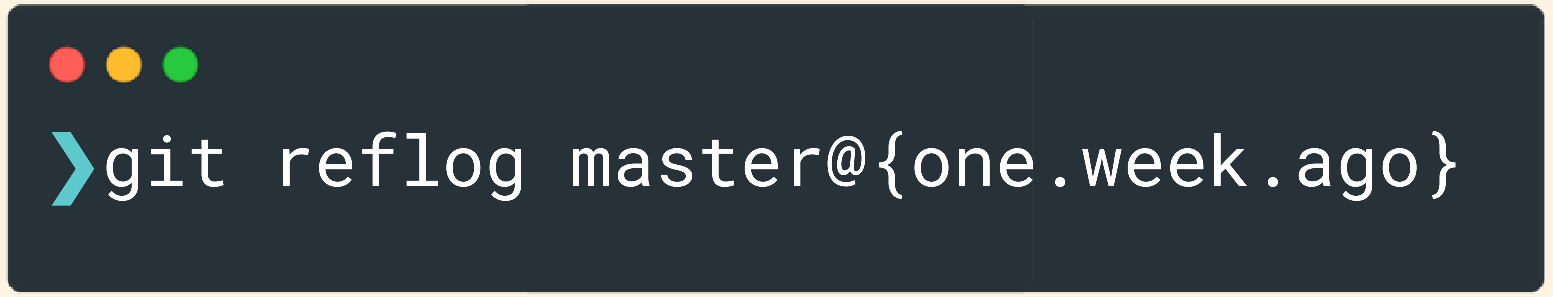


```
name@{qualifier}
```

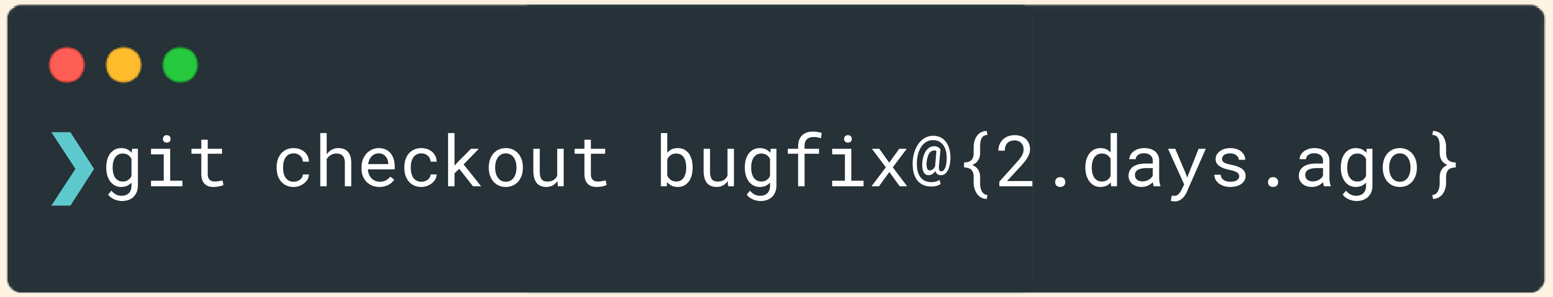
Timed References

Every entry in the reference logs has a timestamp associated with it. We can filter reflogs entries by time/date by using time qualifiers like:

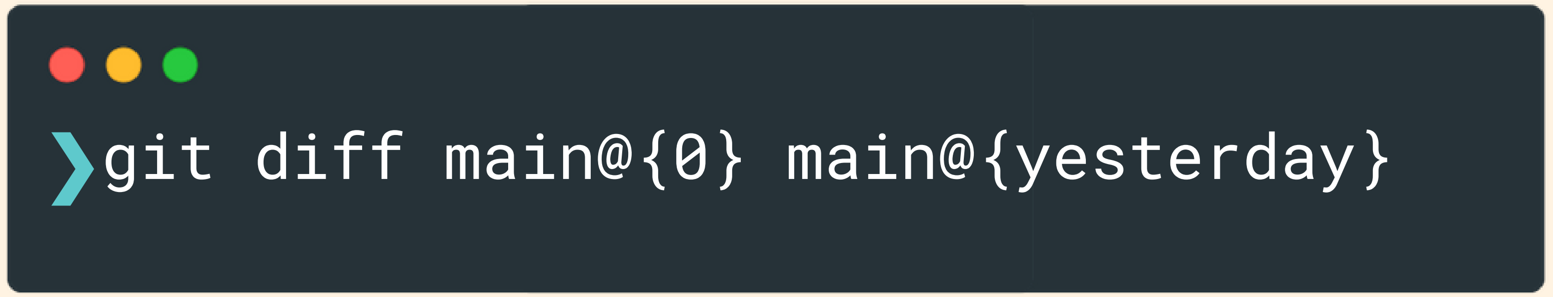
- 1.day.ago
- 3.minutes.ago
- yesterday
- Fri, 12 Feb 2021 14:06:21 -0800



```
❯git reflog master@{one.week.ago}
```



```
❯git checkout bugfix@{2.days.ago}
```



```
❯git diff main@{0} main@{yesterday}
```

Reflogs Rescue

We can sometimes use reflog entries to access commits that seem lost and are not appearing in git log.