



# Git Jungle Guided Tours

#2018--thanos





# Have You Seen My Code?

**Jim Weirich**



# Have You Seen My Code?

Jim Weirich

# Has Anybody Seen My Gal?



parody for git

Marc Peabody



Have You Seen My Code?

Jim Weirich



Has Anybody Seen My Gal?



parody for git

Marc Peabody



parody for bad code

Dean Radcliffe



Have You Seen My Code?

Jim Weirich



Has Anybody Seen My Gal?



parody for git

Marc Peabody



parody for bad code

Dean Radcliffe



Have You Seen My Code?

Jim Weirich

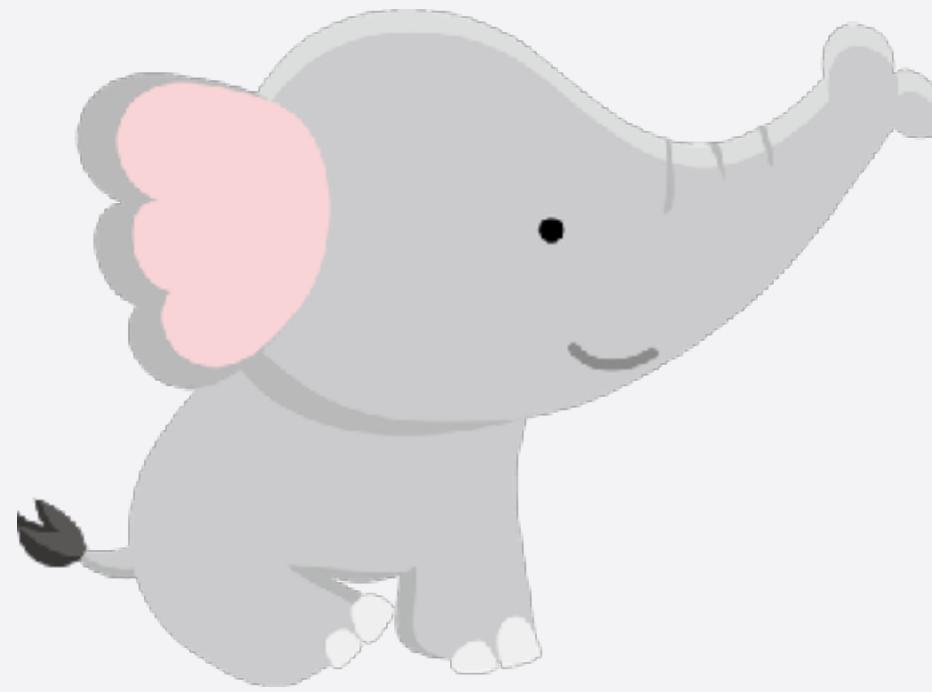
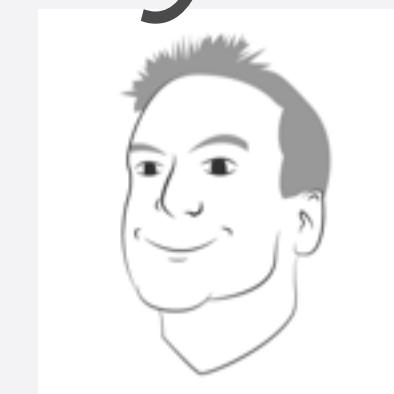


Has Anybody Seen My Gal?



# Marc Peabody

agent #0027



tech: ruby, javascript, java  
other: running, punning, gaming



@testdouble helps teams  
like yours do great things  
[testdouble.com/contact](http://testdouble.com/contact)



Mona



Lisa



git is hard



how  
are  
mistakes  
made?

how  
are  
mistakes  
made?

---

---



how  
are  
mistakes  
made?

---

---



not enough feedback

how  
are  
mistakes  
made?



not enough feedback

how  
are  
mistakes  
made?



not enough feedback

leaky abstractions

how  
are  
mistakes  
made?



not enough feedback

leaky abstractions  
inconsistent API

how  
are  
mistakes  
made?



not enough feedback

leaky abstractions  
inconsistent API  
overloaded commands

how  
are  
mistakes  
made?



not enough feedback

leaky abstractions  
inconsistent API  
overloaded commands

how  
are  
mistakes  
made?



not enough feedback

leaky abstractions  
inconsistent API  
overloaded commands

git --force push

git status

git diff

git log

git fetch





HEAD



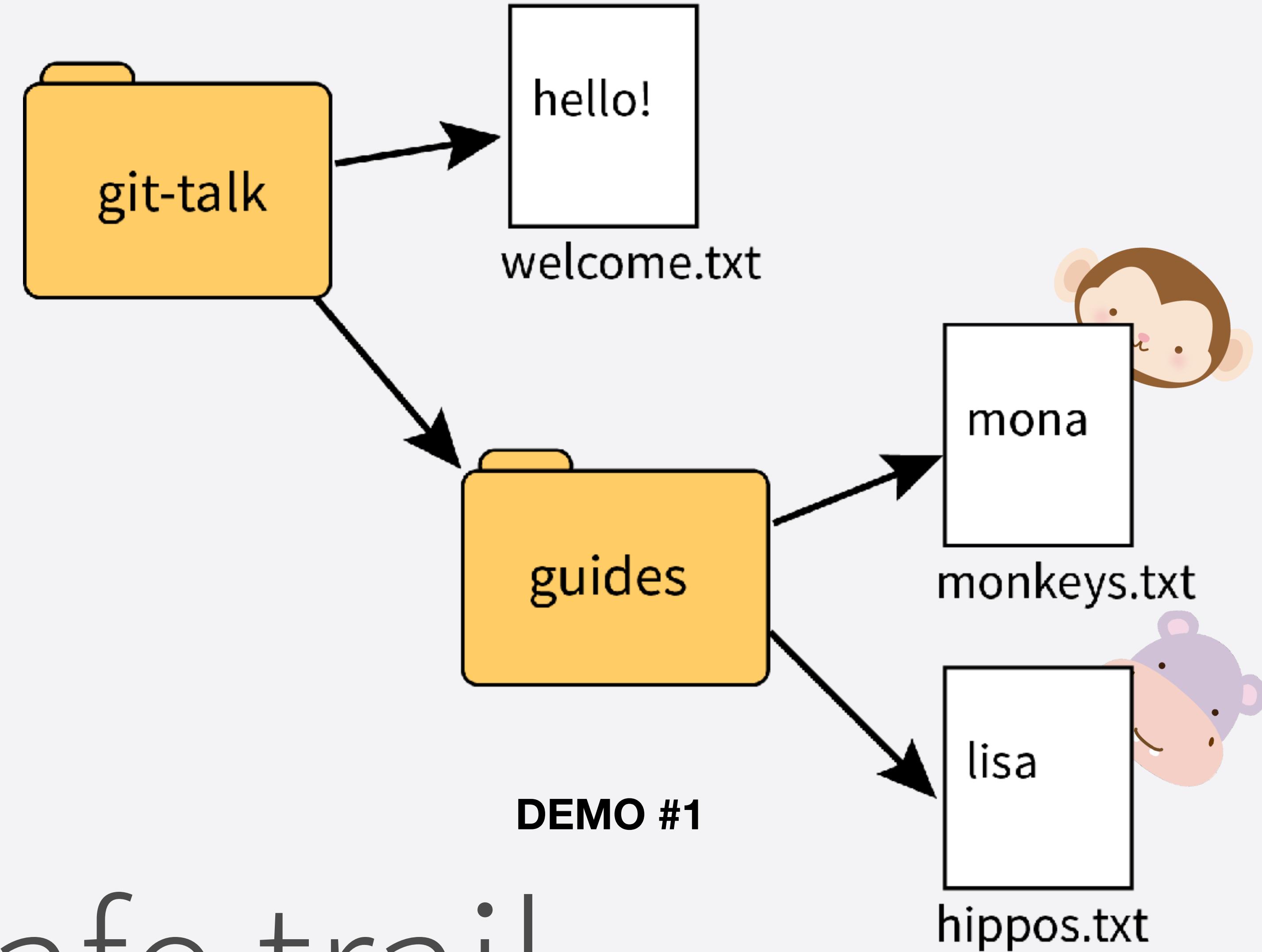
index



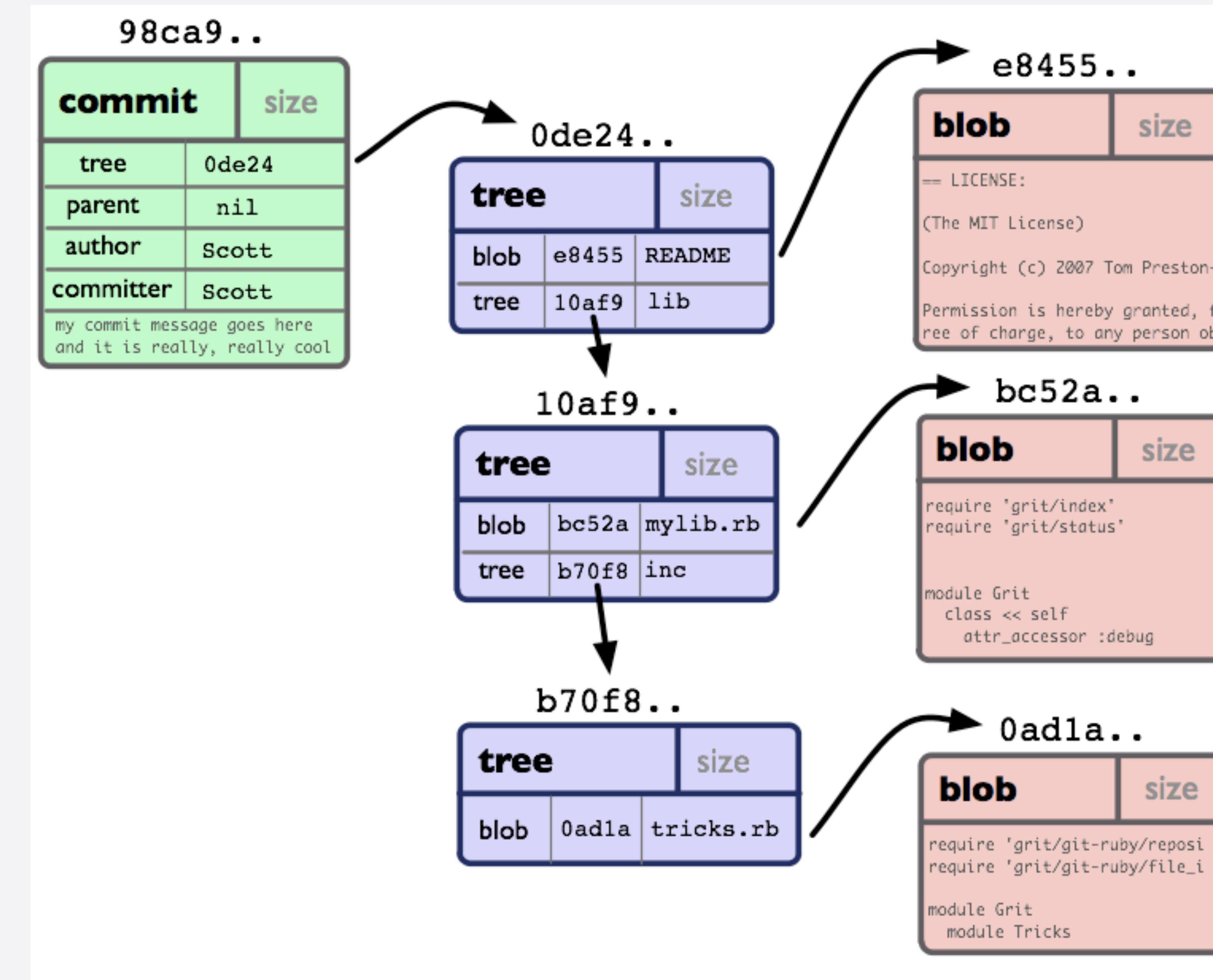
working  
directory



what is a commit?



# the safe trail





# duplicate files

**DEMO #2**

knowing what you know now...

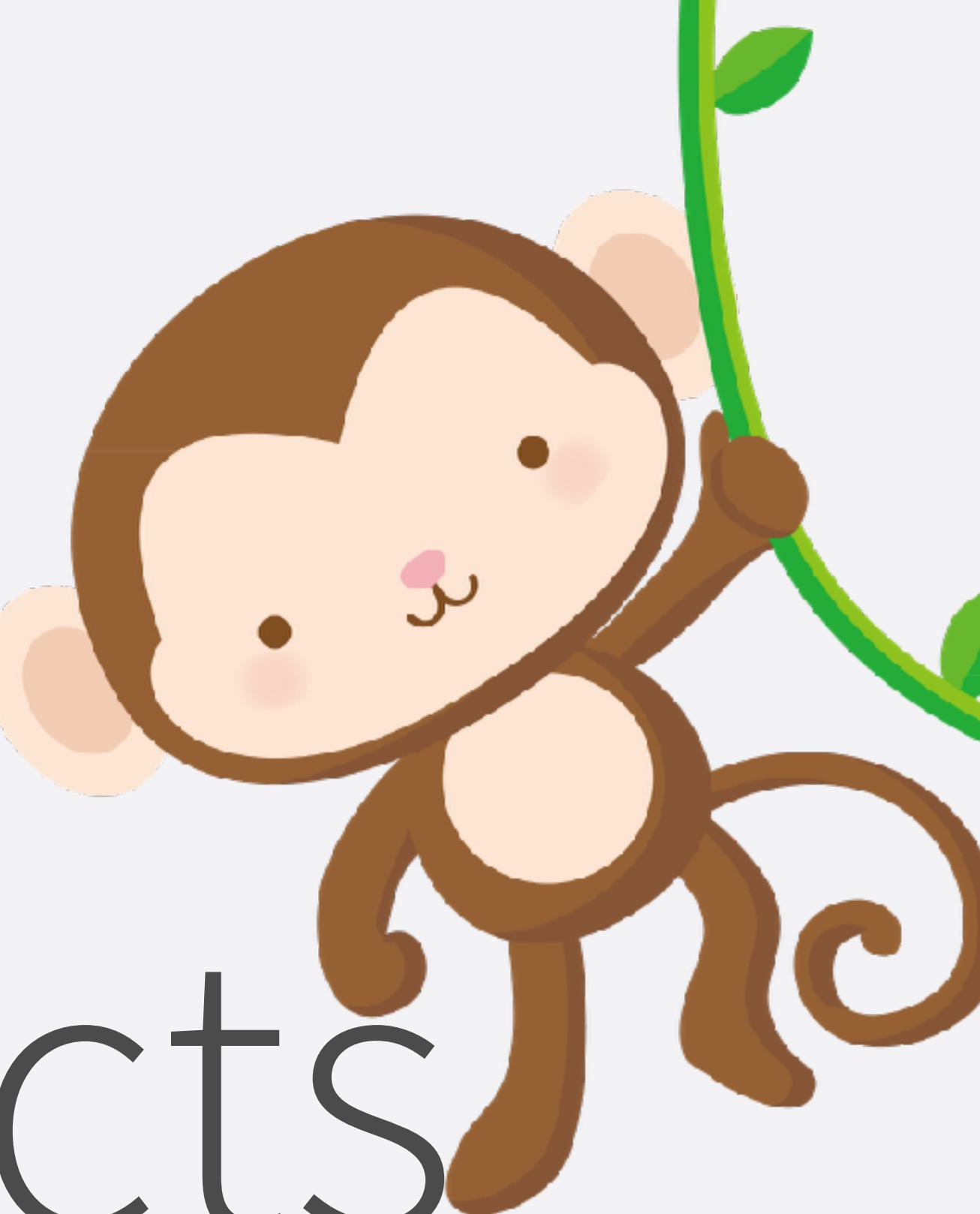


knowing what you know now...



Would my SHAs be the same if I  
ran the same DEMO tomorrow?

# dangling objects





# git object references

**DEMO #3**



# git object references

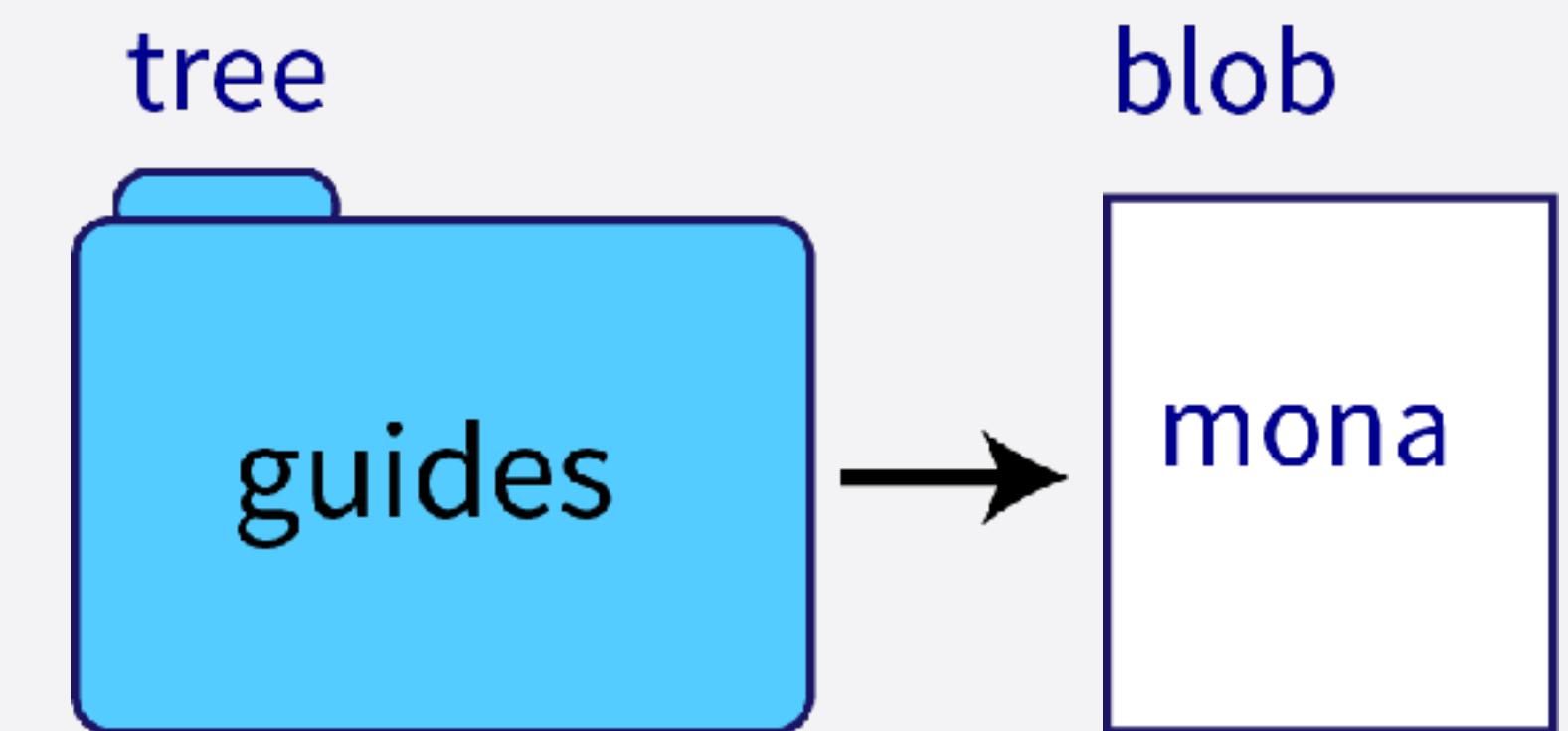
blob



**DEMO #3**



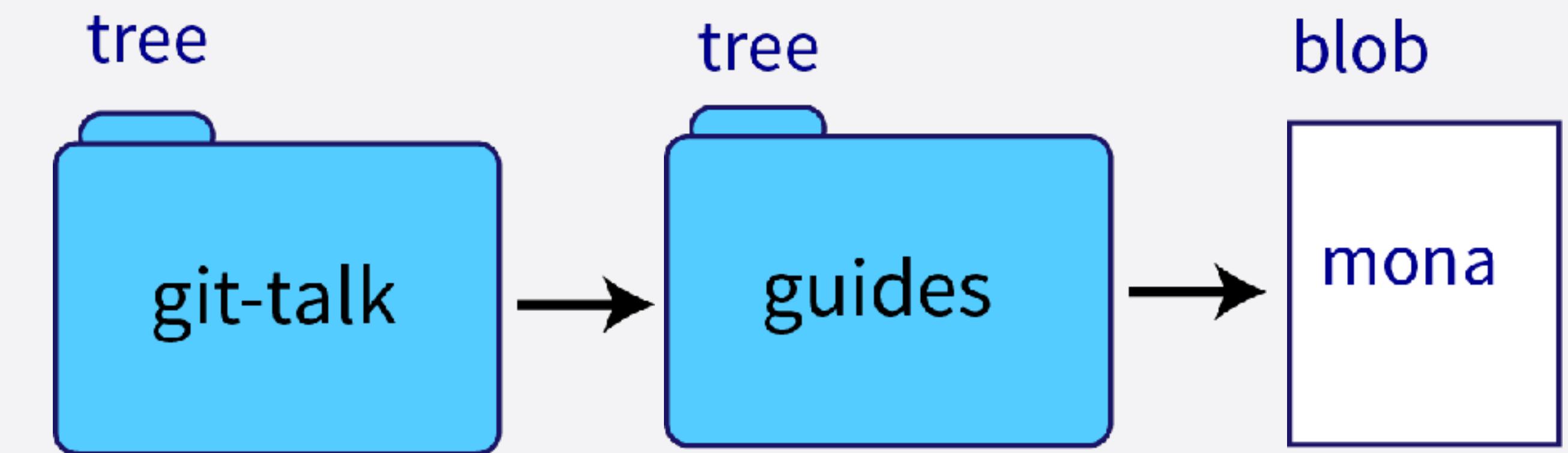
# git object references



**DEMO #3**



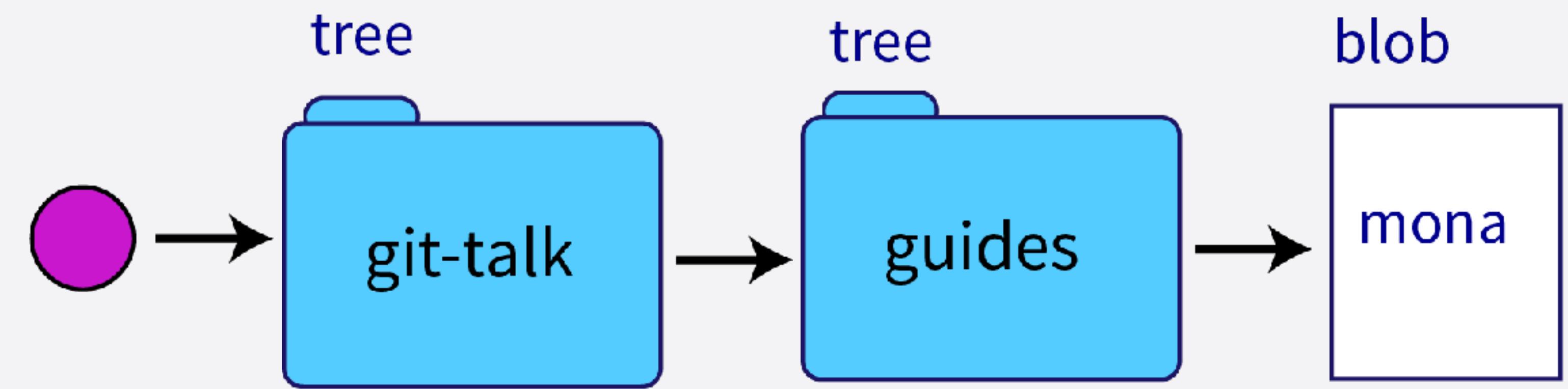
# git object references



**DEMO #3**



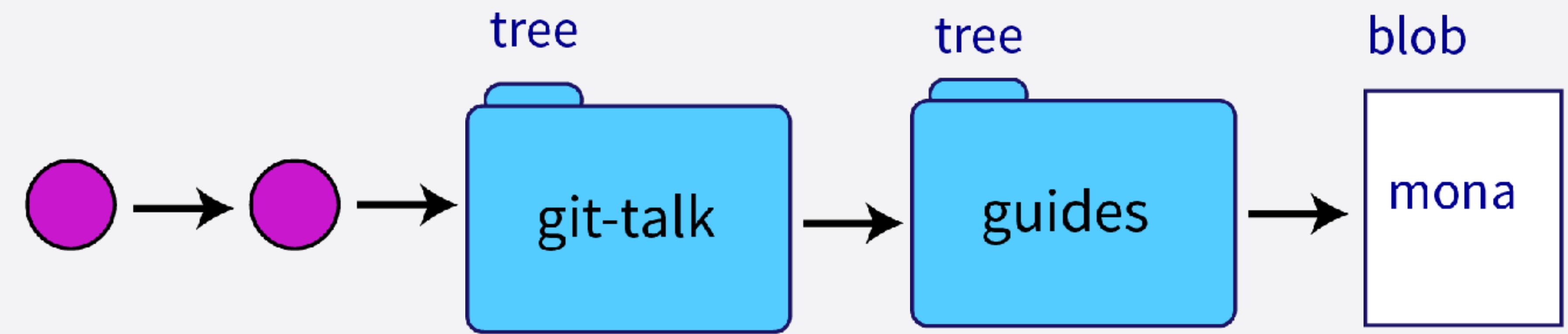
# git object references



**DEMO #3**



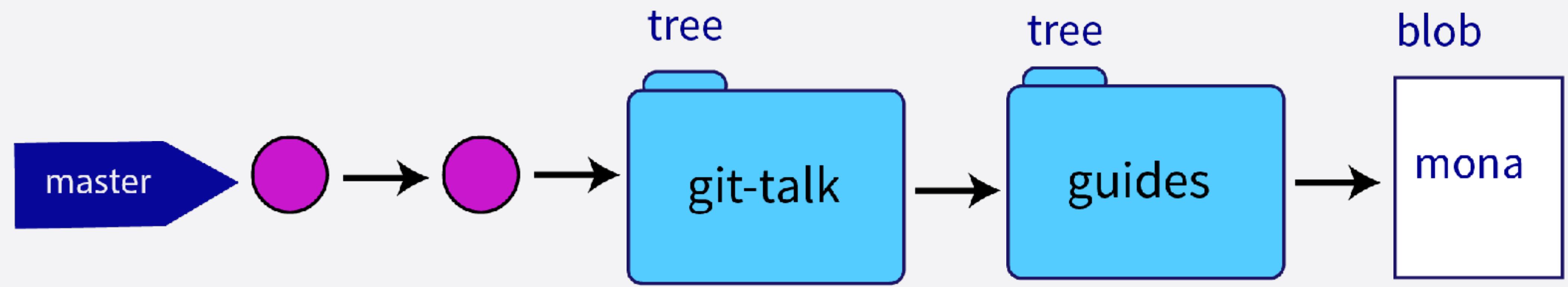
# git object references



**DEMO #3**



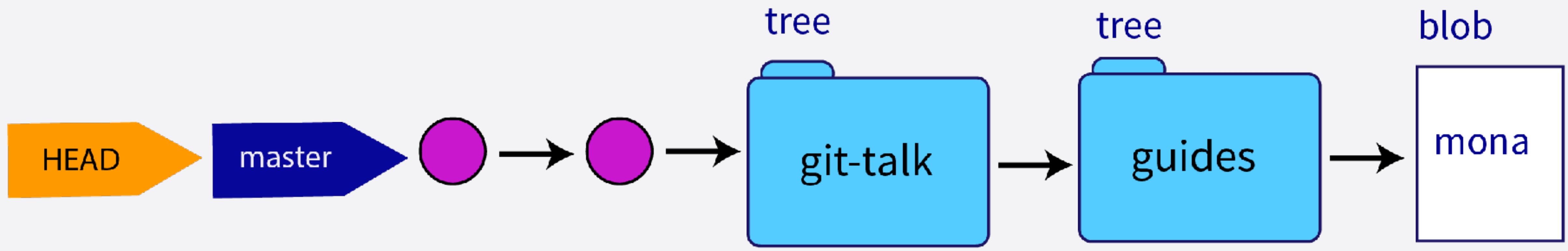
# git object references



**DEMO #3**



# git object references

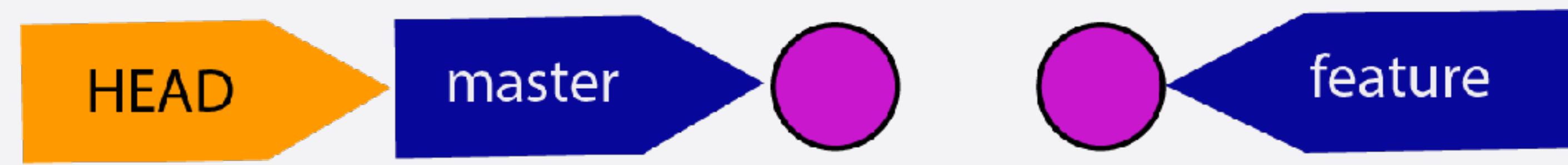


**DEMO #3**

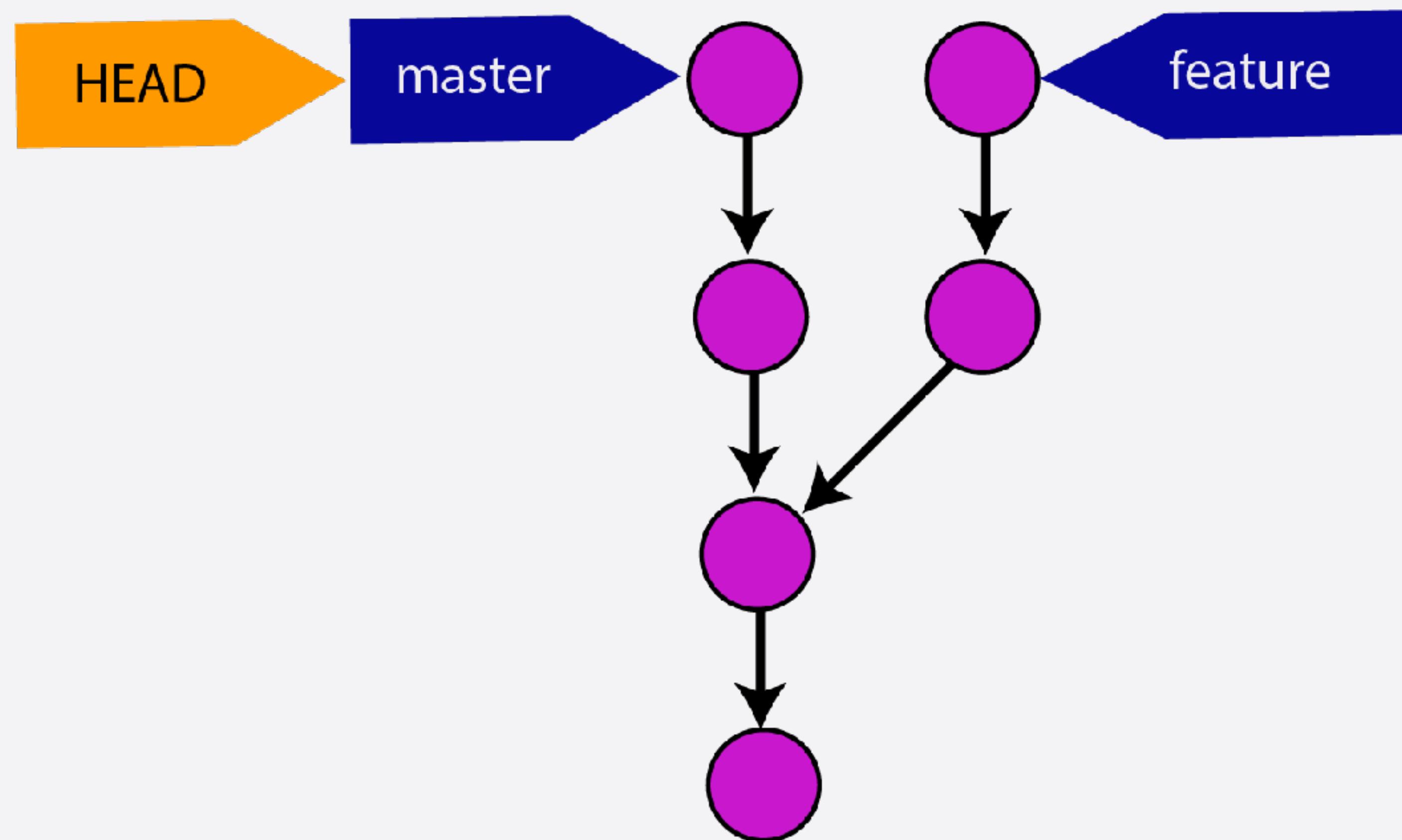
# git log



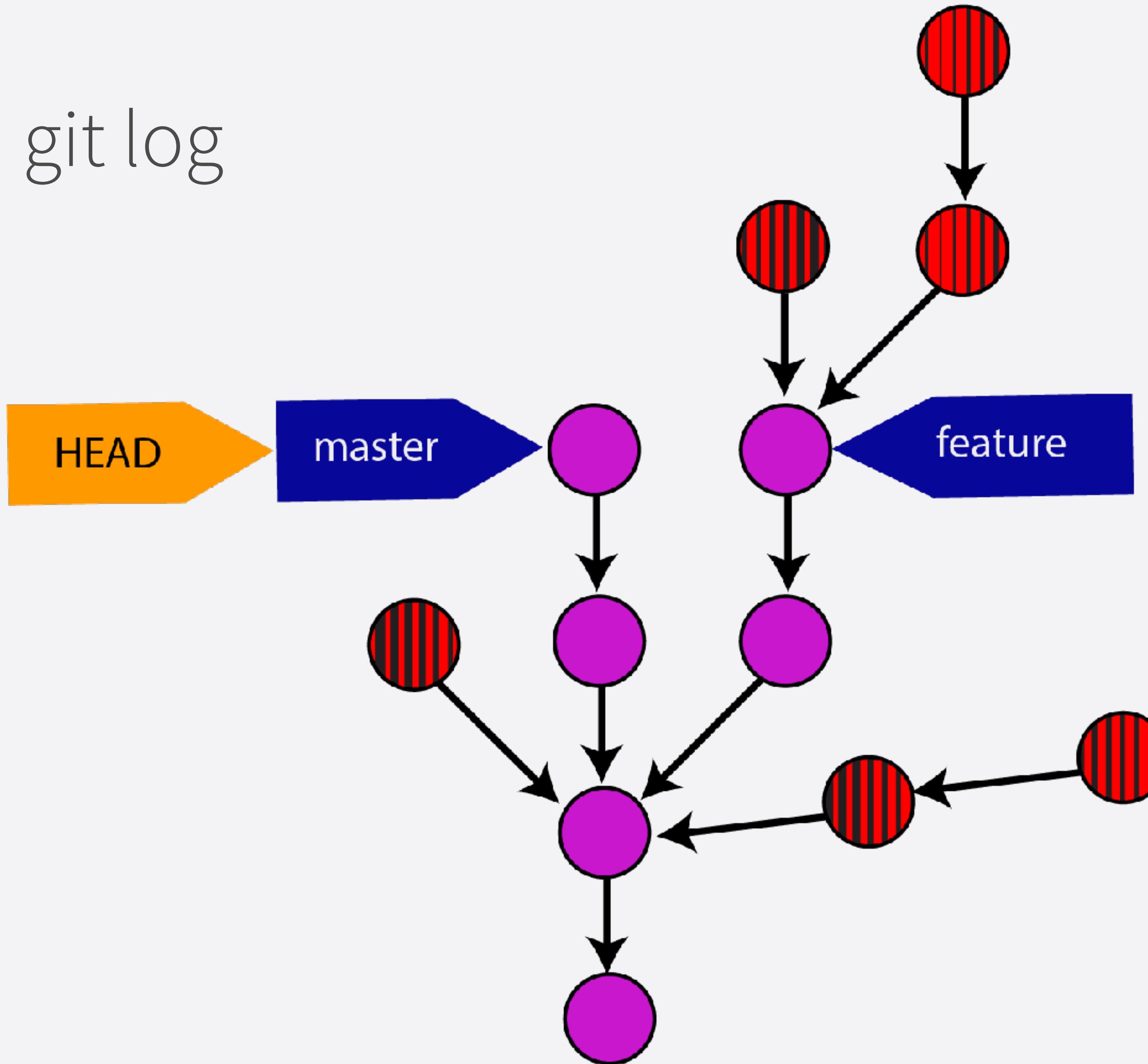
# git log

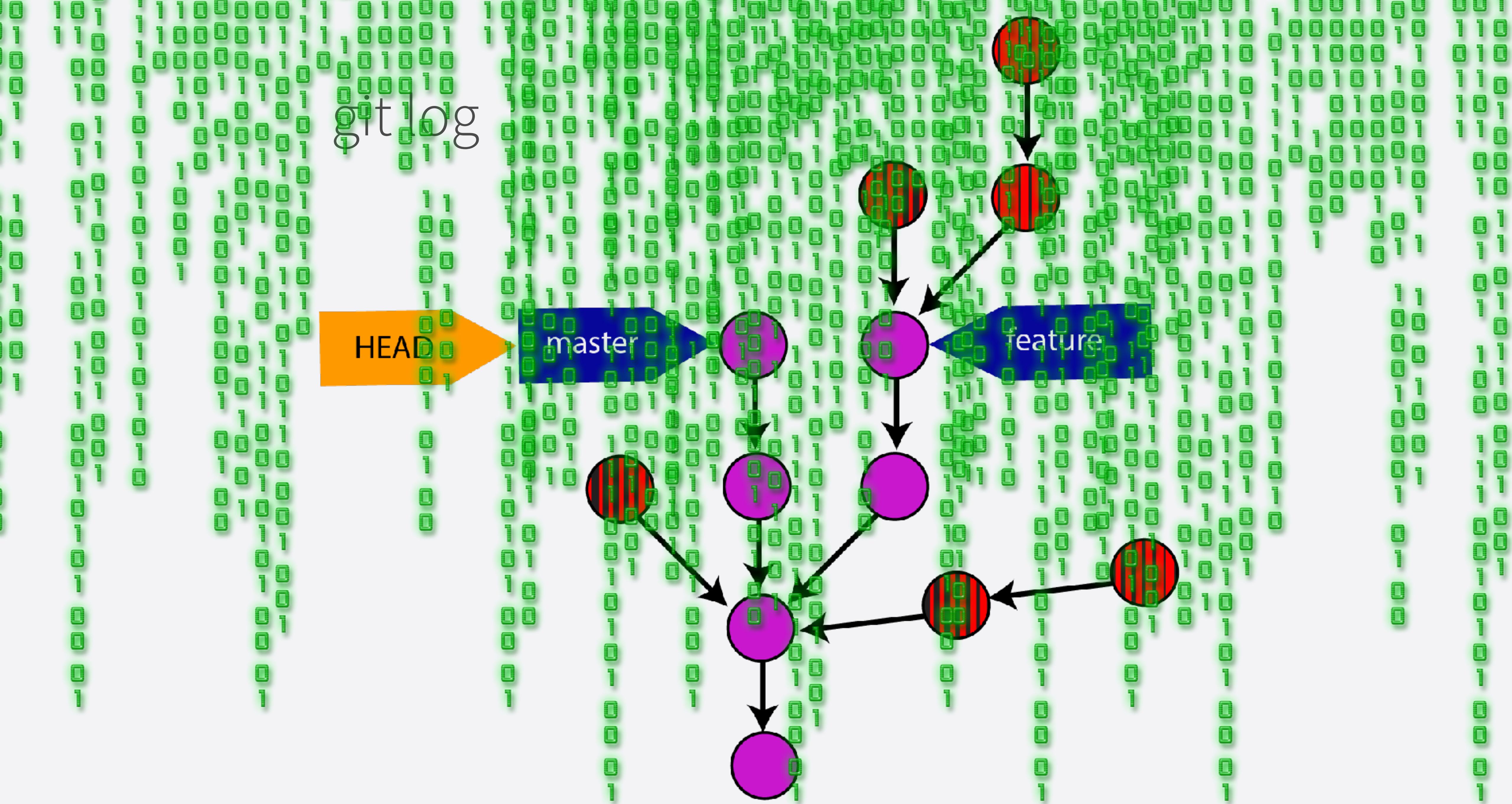


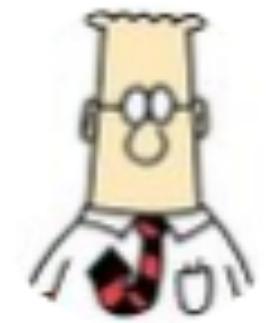
# git log



# git log







# Ranting Engineer

@RantingEngineer



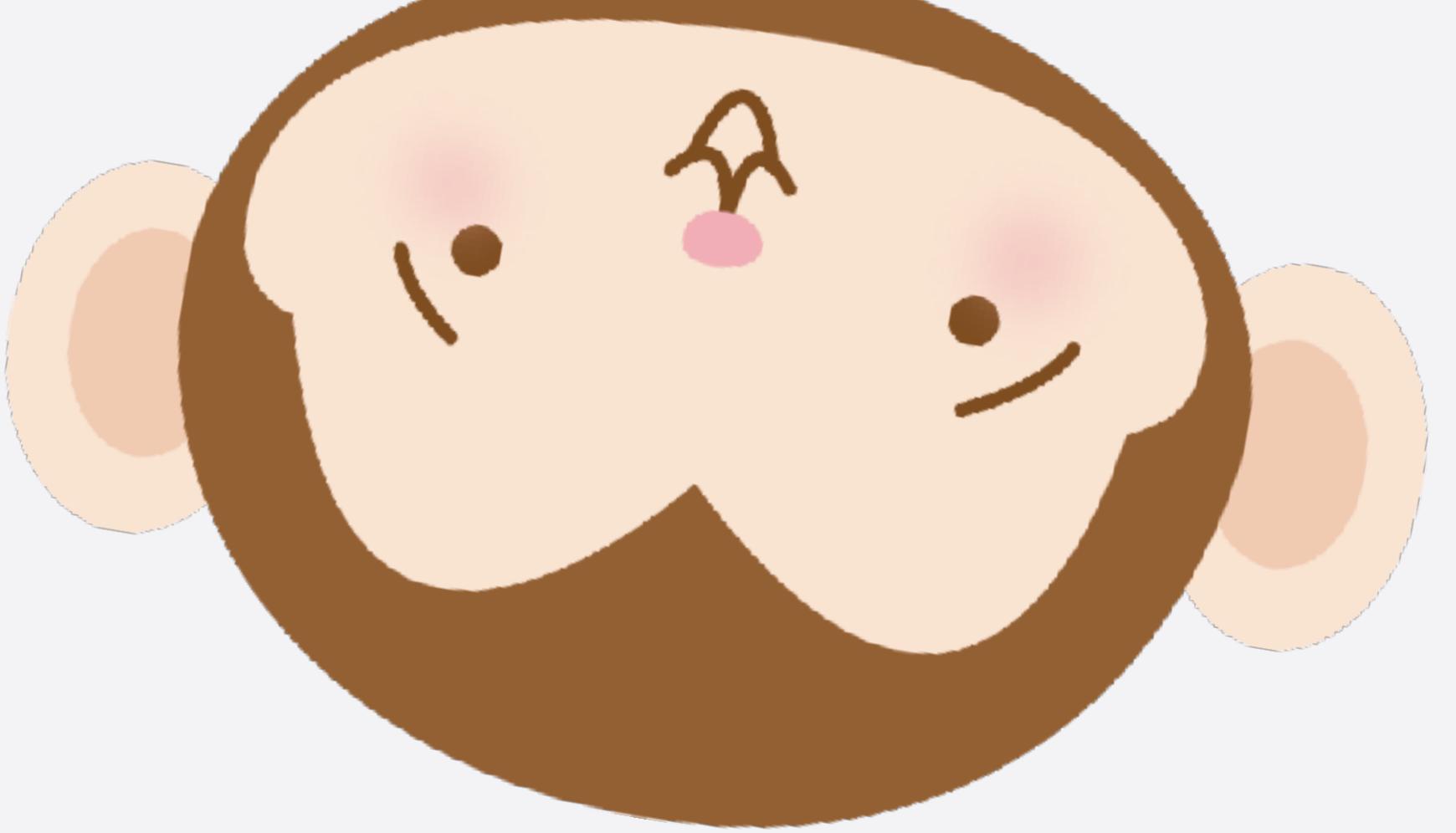
I wonder if I'll ever stop having heart attacks every time I make a mistake in [#git](#)

4:21 PM - Apr 2, 2018



See Ranting Engineer's other Tweets





# recovering lost commits

**DEMO #4**

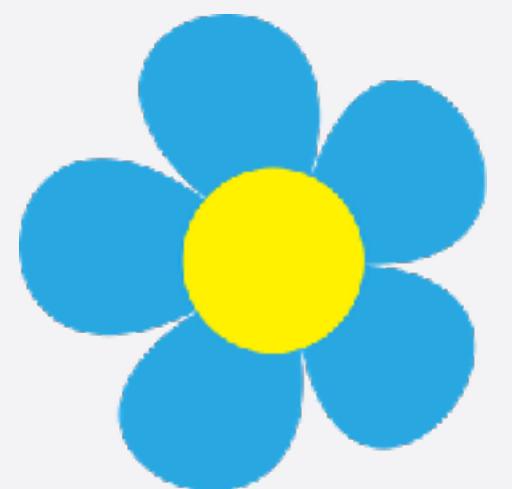
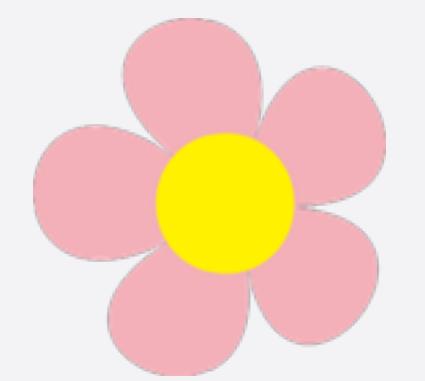
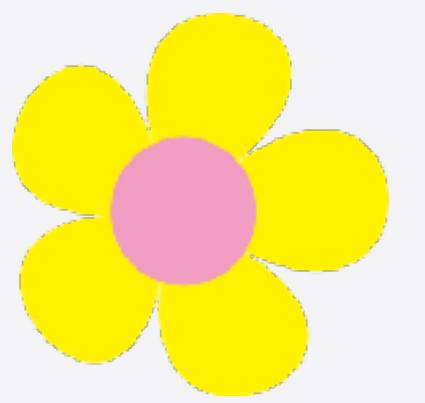


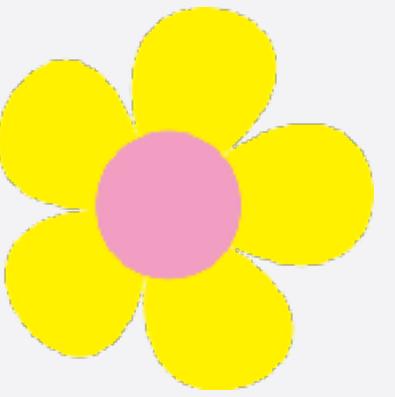
"Oops! I meant to commit  
that to a different branch!"

**DEMO #5**

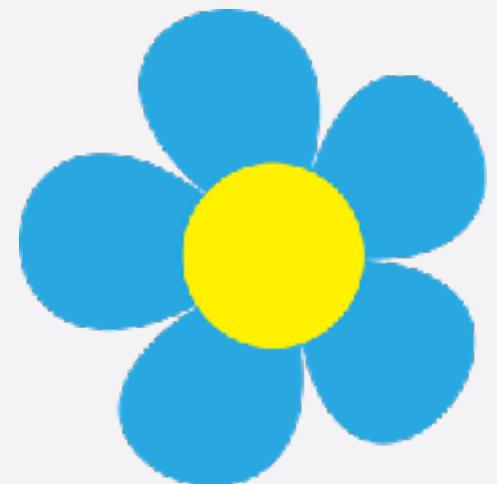
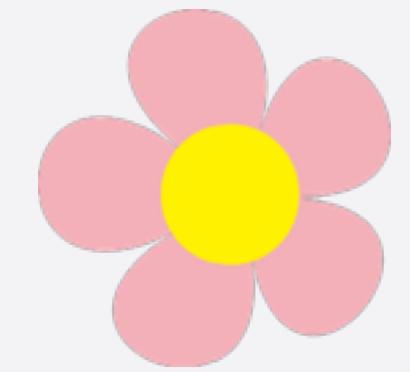
keeping tidy

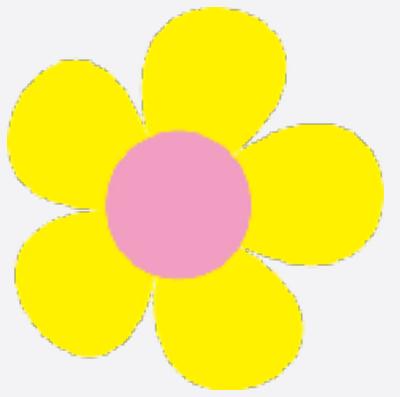






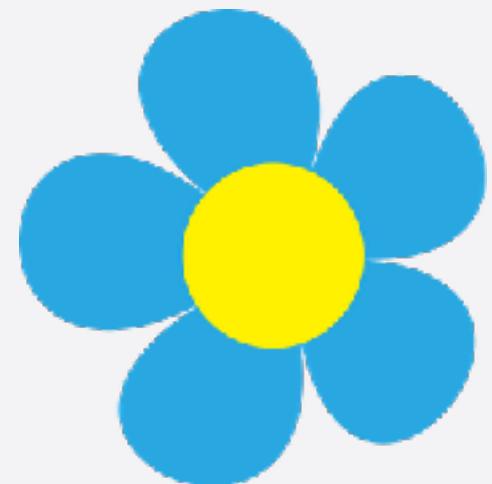
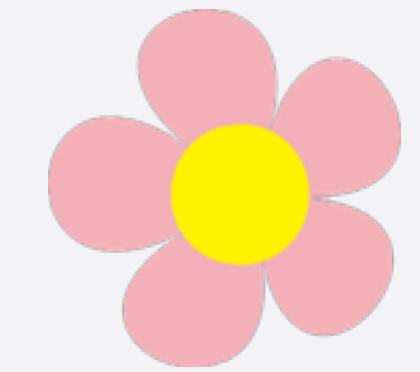
```
git config --global remote.prune true
```

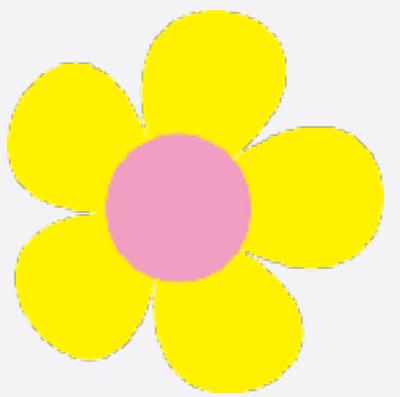




```
git branch -a
```

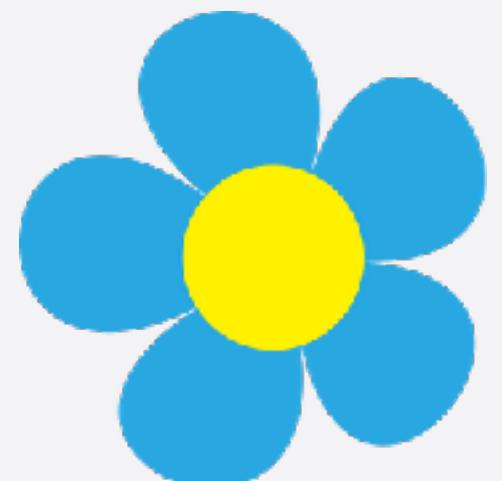
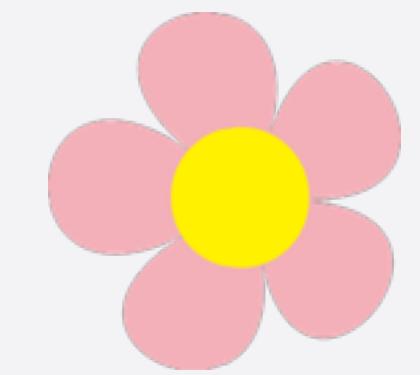
```
git config --global remote.prune true
```





```
git branch -a
```

```
git config --global remote.prune true
```



```
mp/JIRA-1234/api-monkeys-by-tree
```



"Oops! I did not mean  
to delete that branch!"

**DEMO #6**



# merge and rebase

**DEMO #7**





thank you!



#2018--thanos

