

Présenté par



clevy

Introduction à git

Nicolas Coden
Lead Front-End chez Clevy



Qui ne connait pas git ?

(on ne jugera pas)

git pull

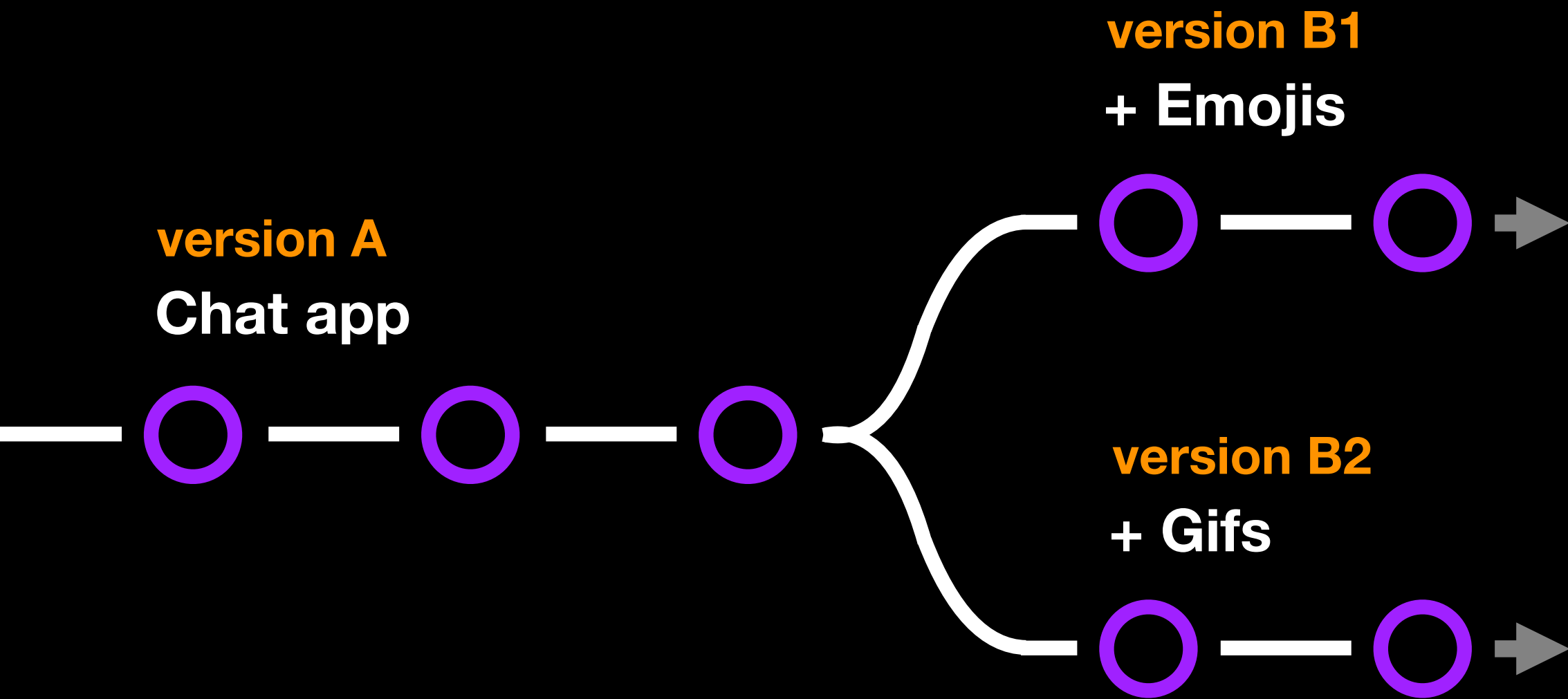
git fetch

git cherry-pick

git reflog

git rebase --interactive



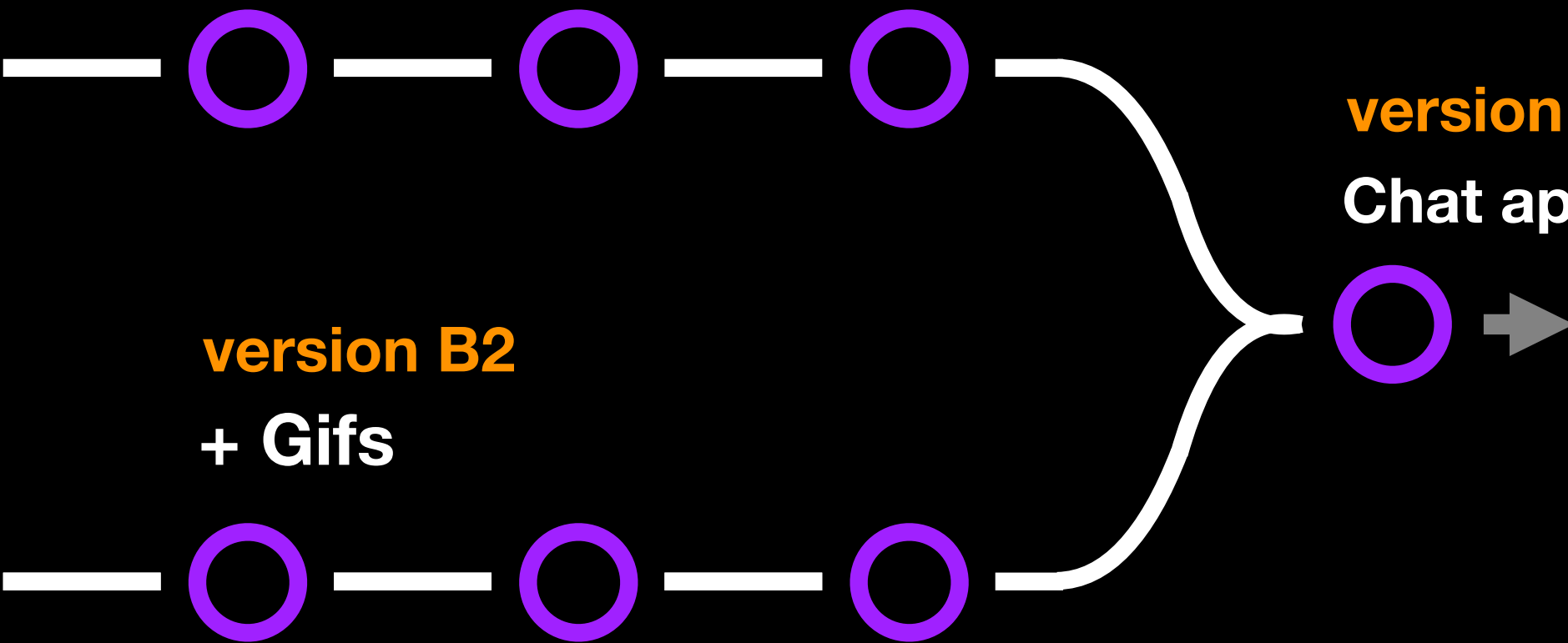


version B1
+ Emojis

version B2
+ Gifs

version C

Chat app with Emojis/Gifs



\$ git clone

\$ git add .

\$ git commit -m “wip”

\$ git push origin master

```
$ git add .
```

```
$ git commit -m "wip"
```

```
$ git push origin master
```

```
Error: failed to push to ma...
```

\$ git push origin master

Error: failed to push to ma...

\$ git status

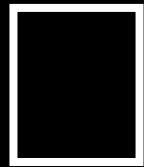
CRITICAL: will destroy hu...

fuck this

\$ rm -rf *project*

\$ git clone

Files



core.c



lib.c



msg.c



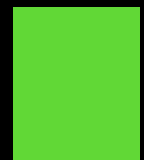
emote.c



emoji-1.c



emoji-2.c



emoji-3.c



emoji-4.c

 Inchangé

 Ajouté

 Modifié

 Supprimé

git add

git commit

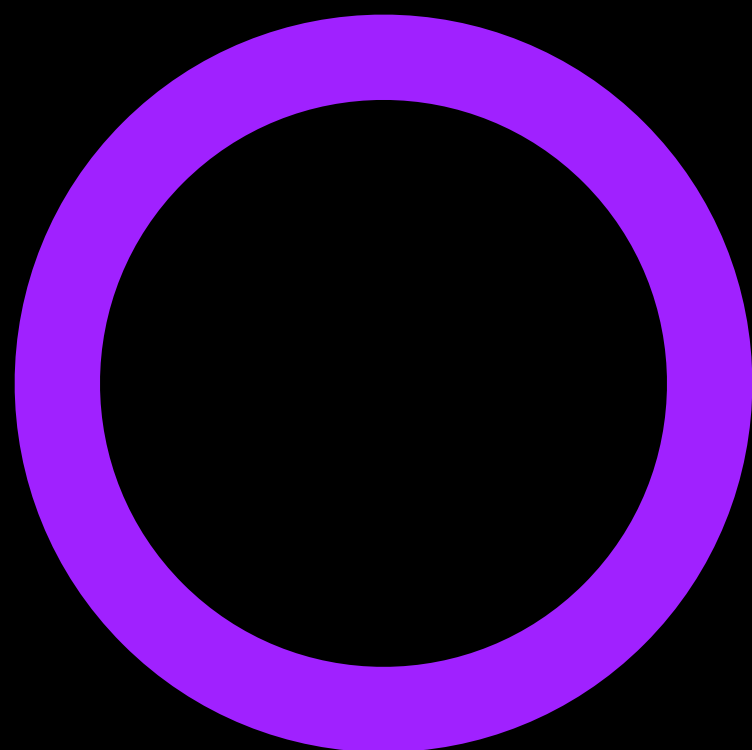


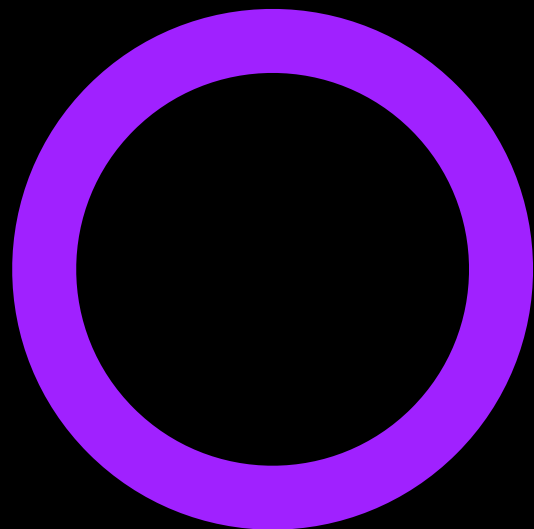
git reset

■ Inchangé ■ Ajouté ■ Modifié ■ Supprimé



What's inside our commit





+ 39 lines added in A

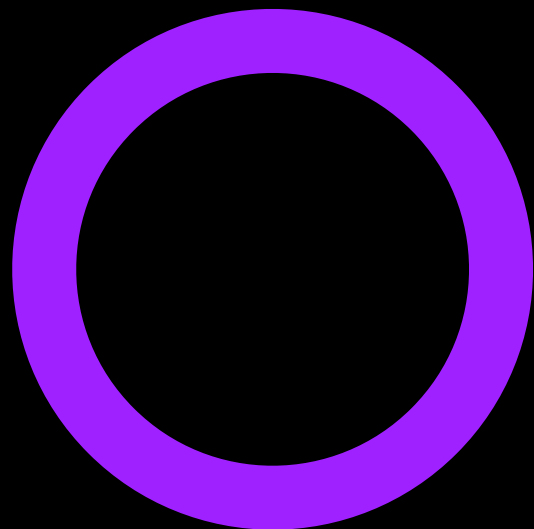
+ 6 lines added in B

- 9 lines removed in Y

+ File X added

+ File Y added

- File Z removed



e11baf7fe9b1f

+ 39 lines added in A

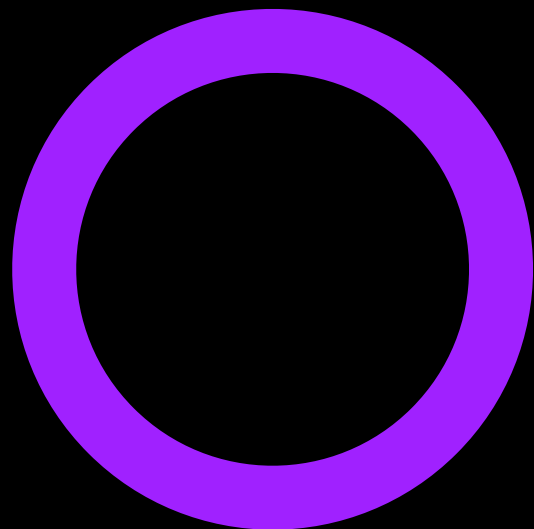
+ 6 lines added in B

- 9 lines removed in Y

+ File X added

+ File Y added

- File Z removed



e11baf7fe9b1f

Committed by Nicolas
At 15:57 01/03/19

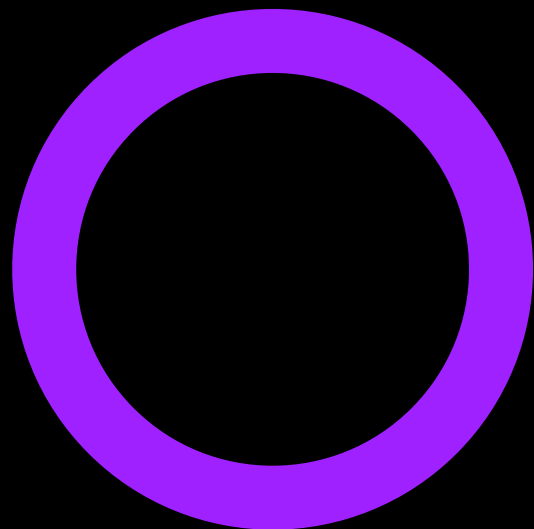
+ 39 lines added in A

+ 6 lines added in B

- 9 lines removed in Y

+ File X added

+ File Y added



e11baf7fe9b1f

○ Parent: ae9b4f90eebaf5

Committed by Nicolas

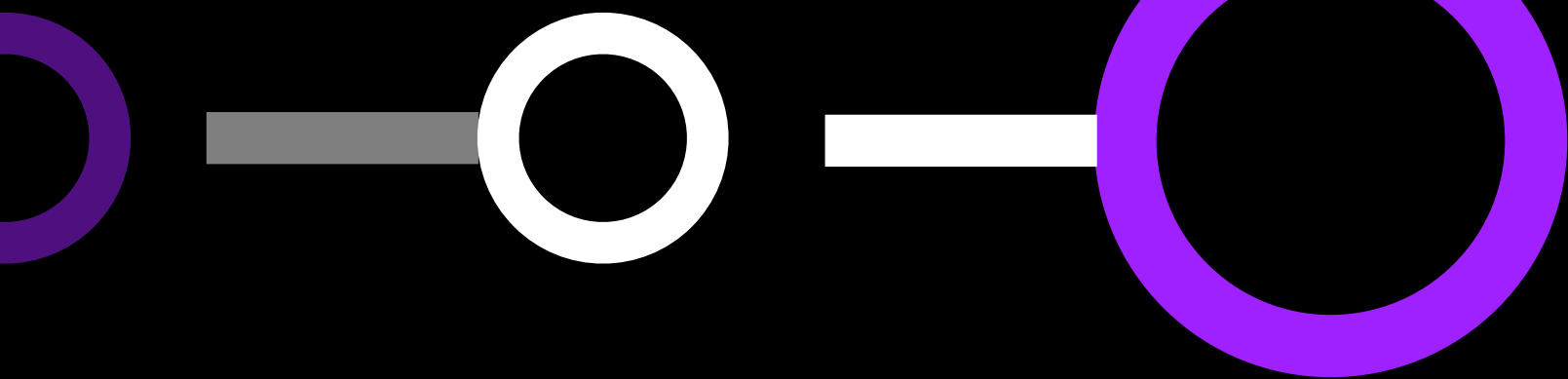
At 15:57 01/03/19

+ 39 lines added in A

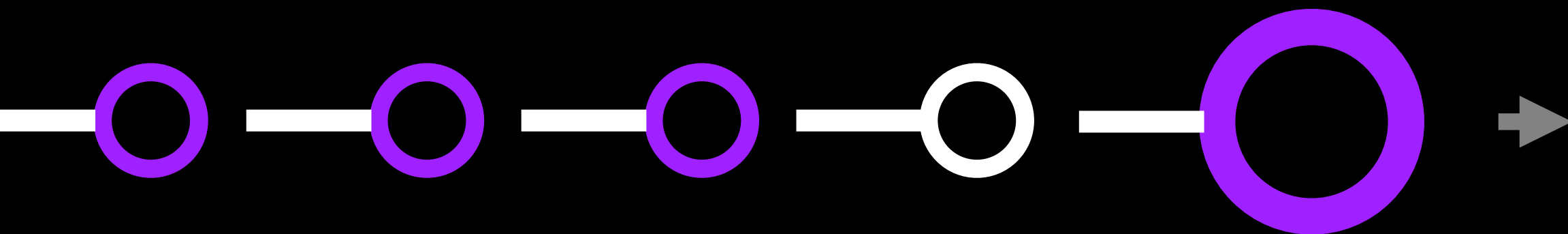
+ 6 lines added in B

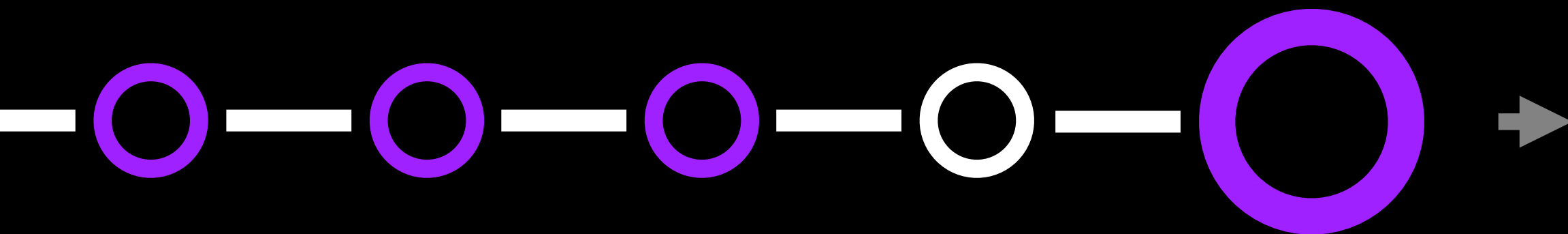
- 9 lines removed in Y

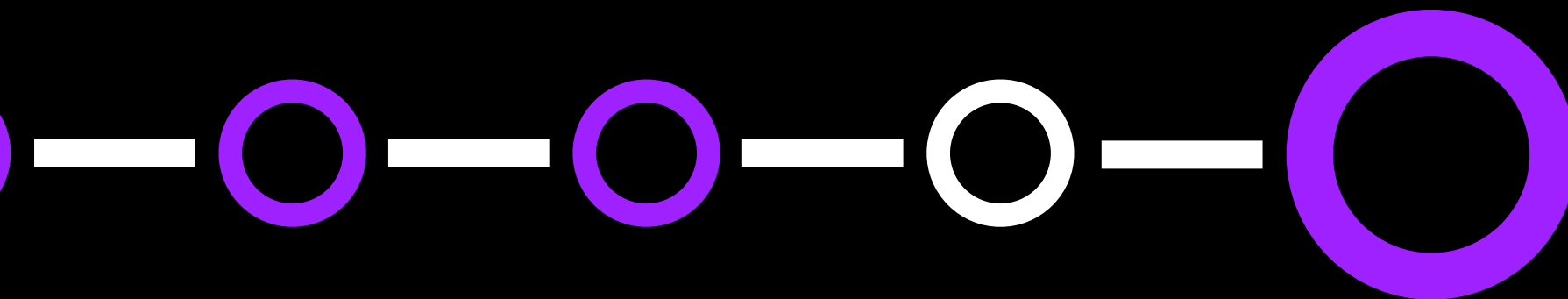
+ File X added



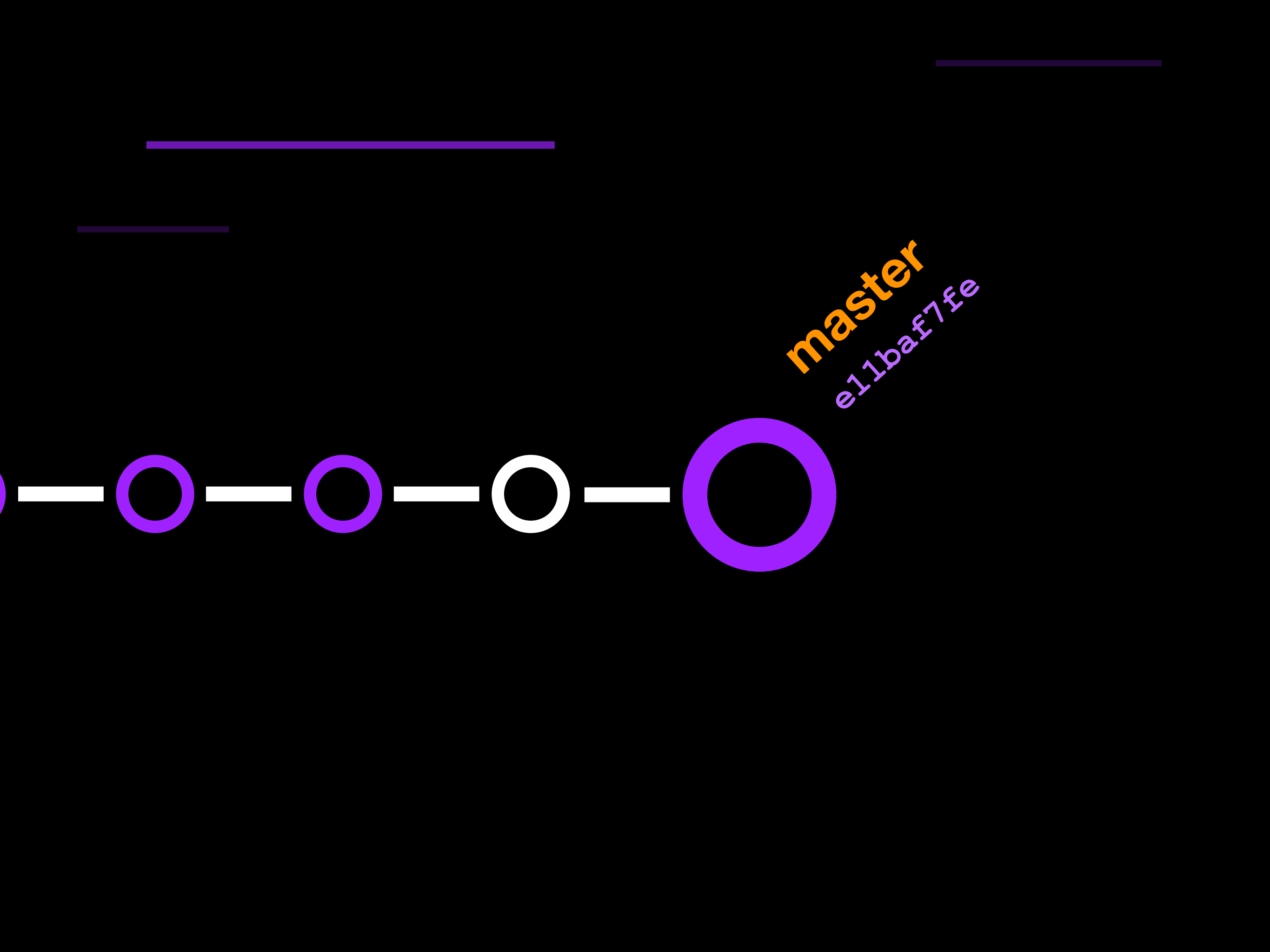
Branches







master



master

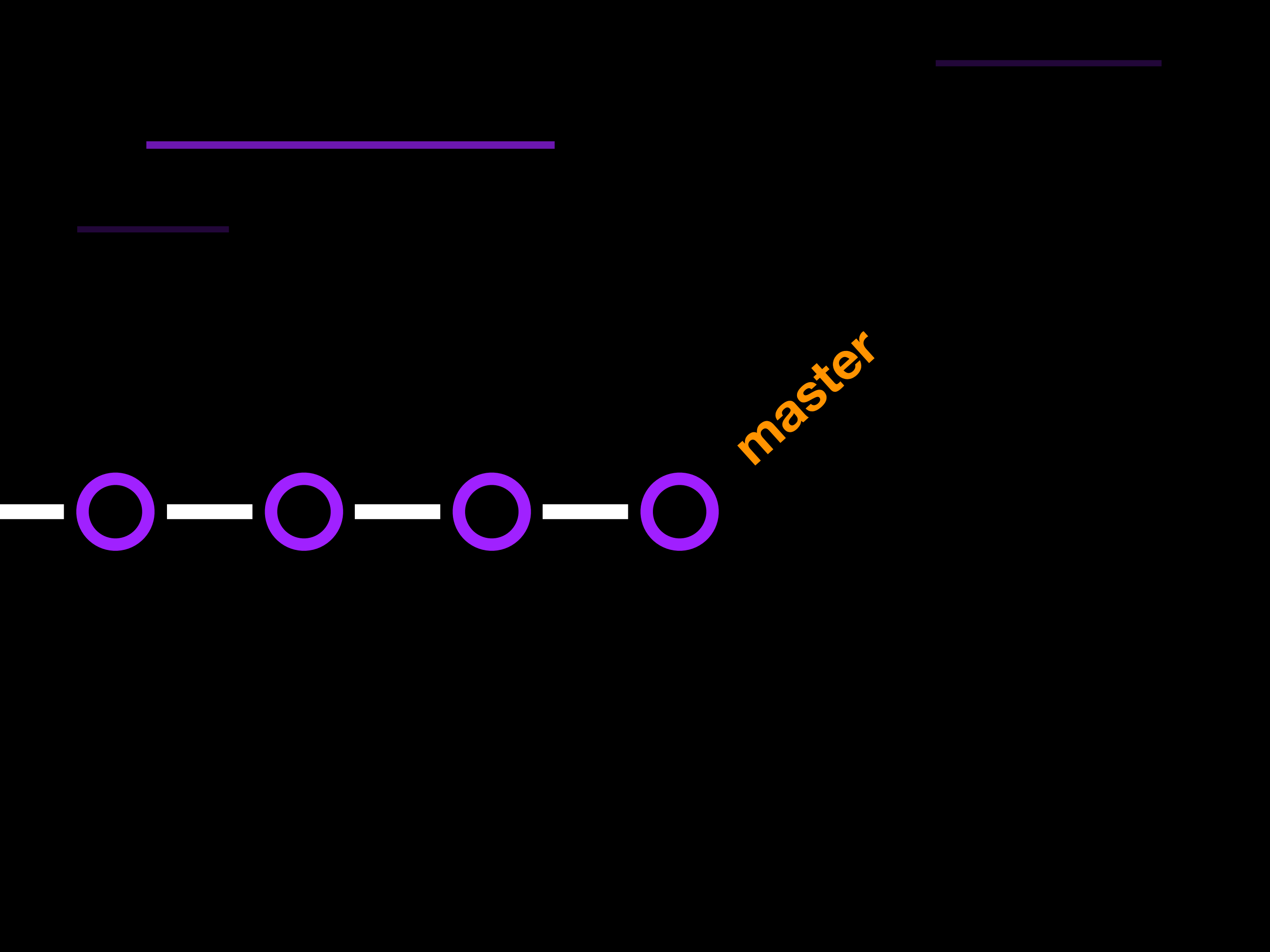
e11baf7fe9b1f

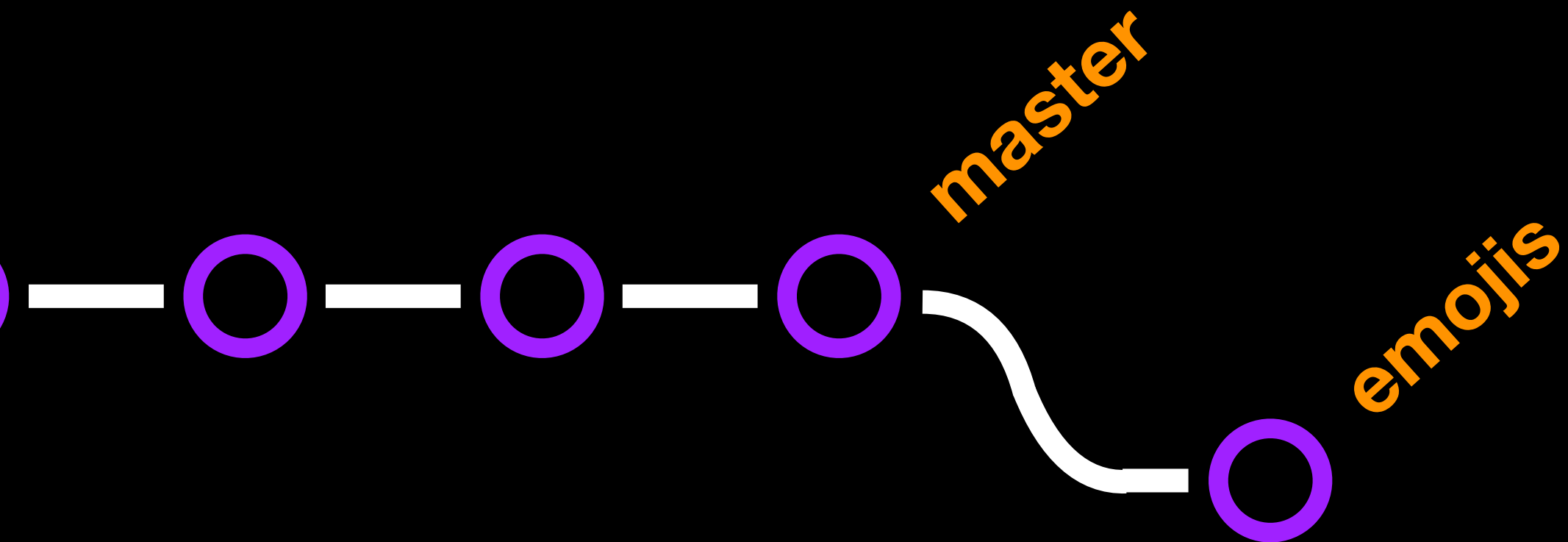
develop

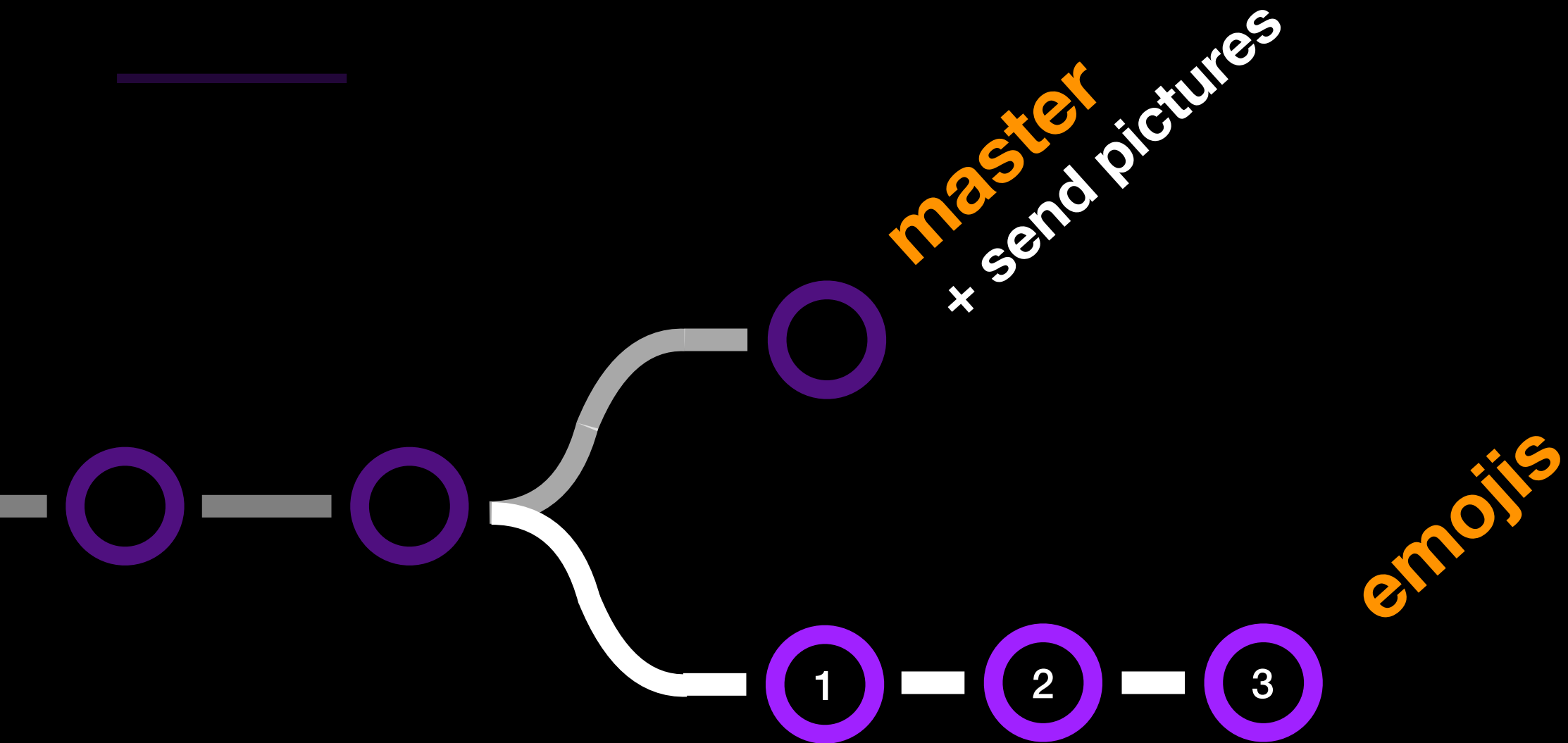
67bfe9af7b1fe

beta

099b1fecc7fe9







\$ git merge

\$ git cherry-pick

\$ git rebase

\$ copy & paste

Copy & Paste

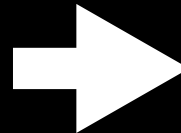
feature

```
#include <stdio.h>
int main(){
    int dividend;

    printf("Enter dividend: ");
    expected = !sp_inquisition;
    scanf("%d", &dividend);
    temp = dividend + 1;

    printf("Enter divisor: ");

    remainder = dividend;
    yourenot = payingattention;
    printf("Quotient = %d\n)
    printf("I loose the game");
    printf(temp);
}
```



master

```
#include <stdio.h>
int main(){
    int dividend;

    scanf("%d", &dividend);
    temp = dividend + 1;

    printf("Enter divisor: ");

    remainder = dividend;
    yourenot = payingattention;
    printf("Quotient = %d\n)
    printf("I loose the game");
    printf(temp);
}
```

Copy & Paste

feature

```
#include <stdio.h>
int main(){
    int dividend;

    printf("Enter dividend: ");
    expected = !sp_inquisition;
    scanf("%d", &dividend);
    temp = dividend + 1;

    printf("Enter divisor: ");
    printf("Do not enter 0");

    remainder = dividend;
    yourenot = payingattention;
    printf("Quotient = %d\n");
    printf("I loose the game");
}
```



master

```
#include <stdio.h>
int main(){
    int dividend;

    scanf("%d", &dividend);
    scanf("%d", &dividend * 2);
    divisor = !dividend - rem;
    temp = dividend + 1;

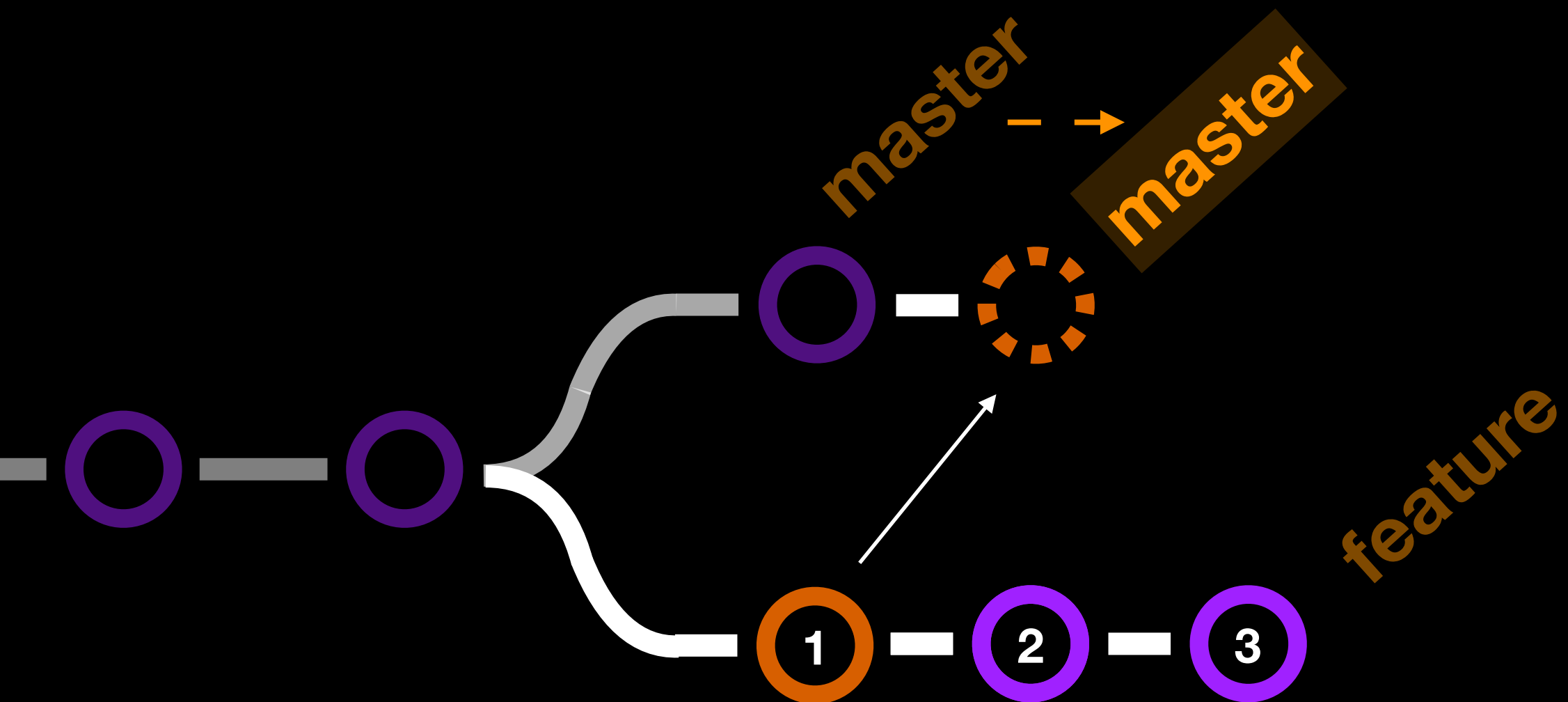
    printf("Enter divisor: ");

    remainder = dividend;
    yourenot = payingattention;
    printf("Quotient = %d\n");
    printf("I loose the game");
}
```


That's conflicts

*When git does not know how to resolve changes
between two versions.*

\$ git cherry-pick 1



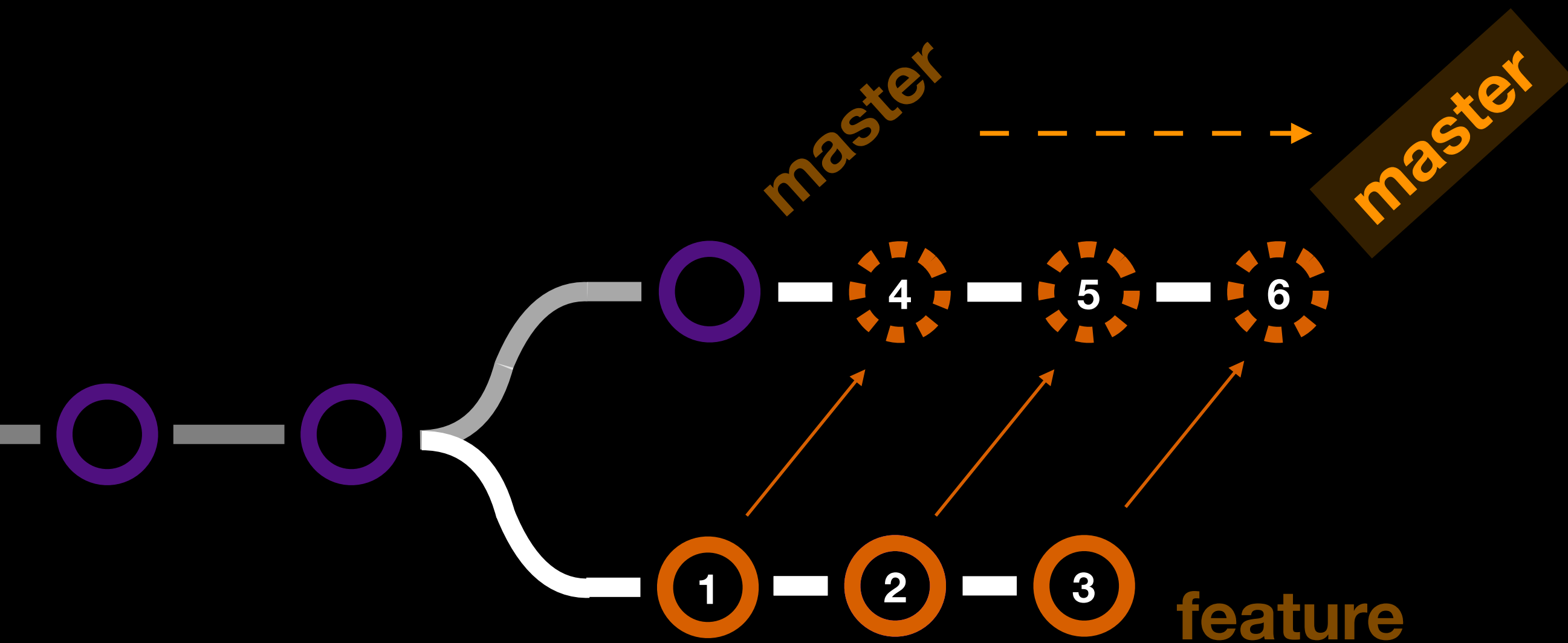


```
git cherry-pick <hash>
```

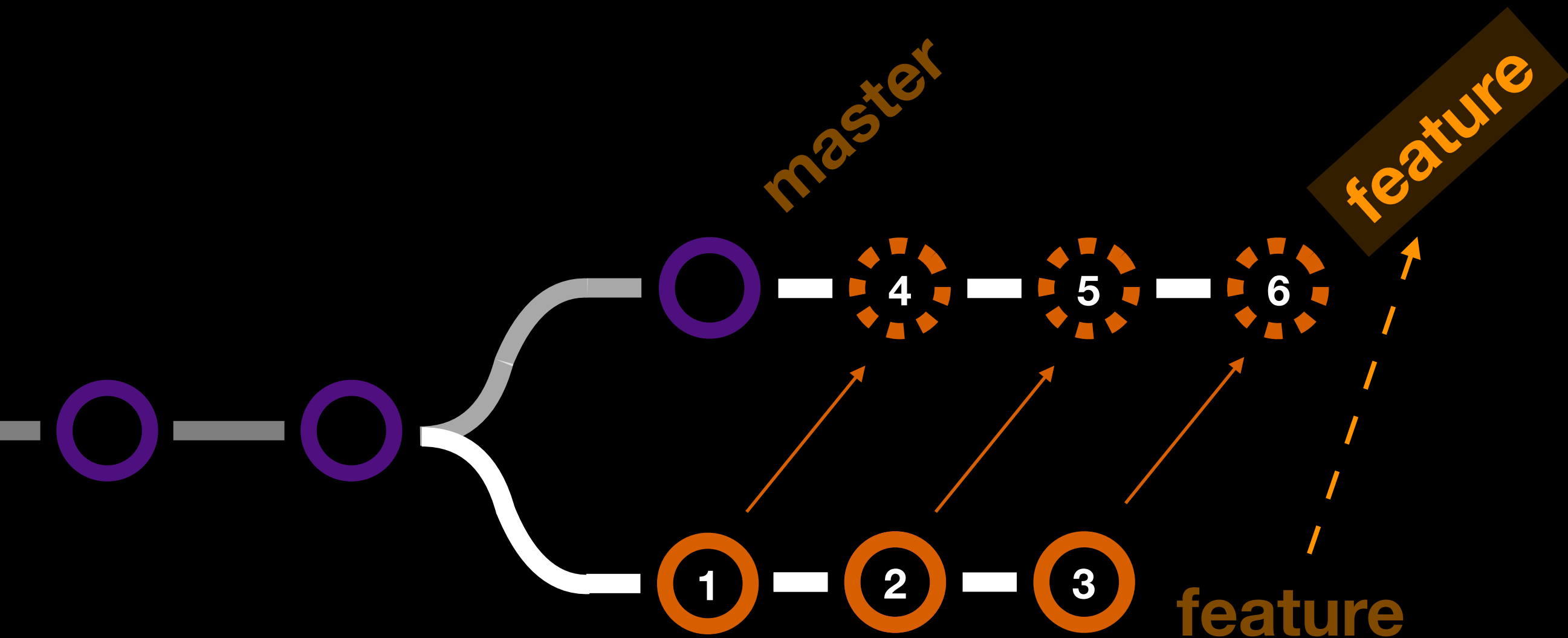
Copy the commit <hash> into the current branch.

Example: integrate a fix that was made in the wrong branch.

\$ git cherry-pick (x3)



\$ git rebase master



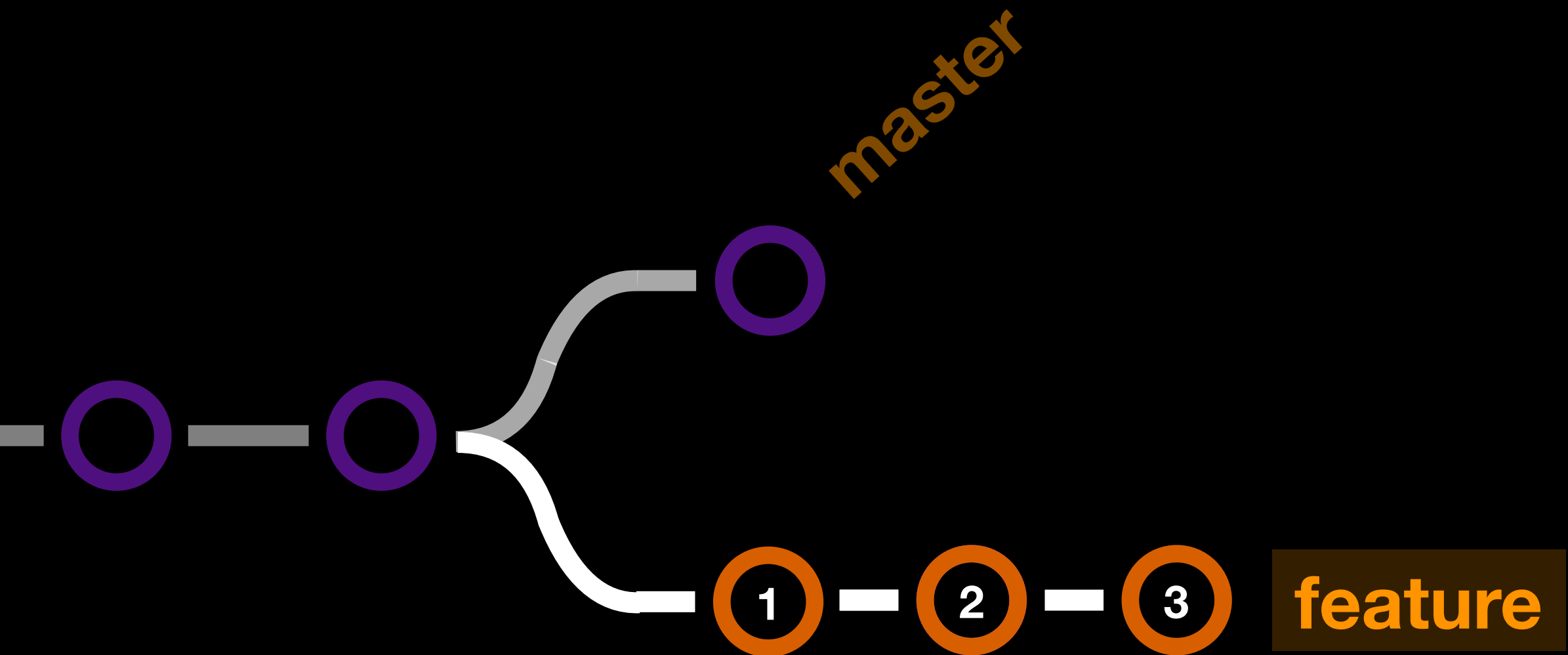


```
git rebase <branch>
```

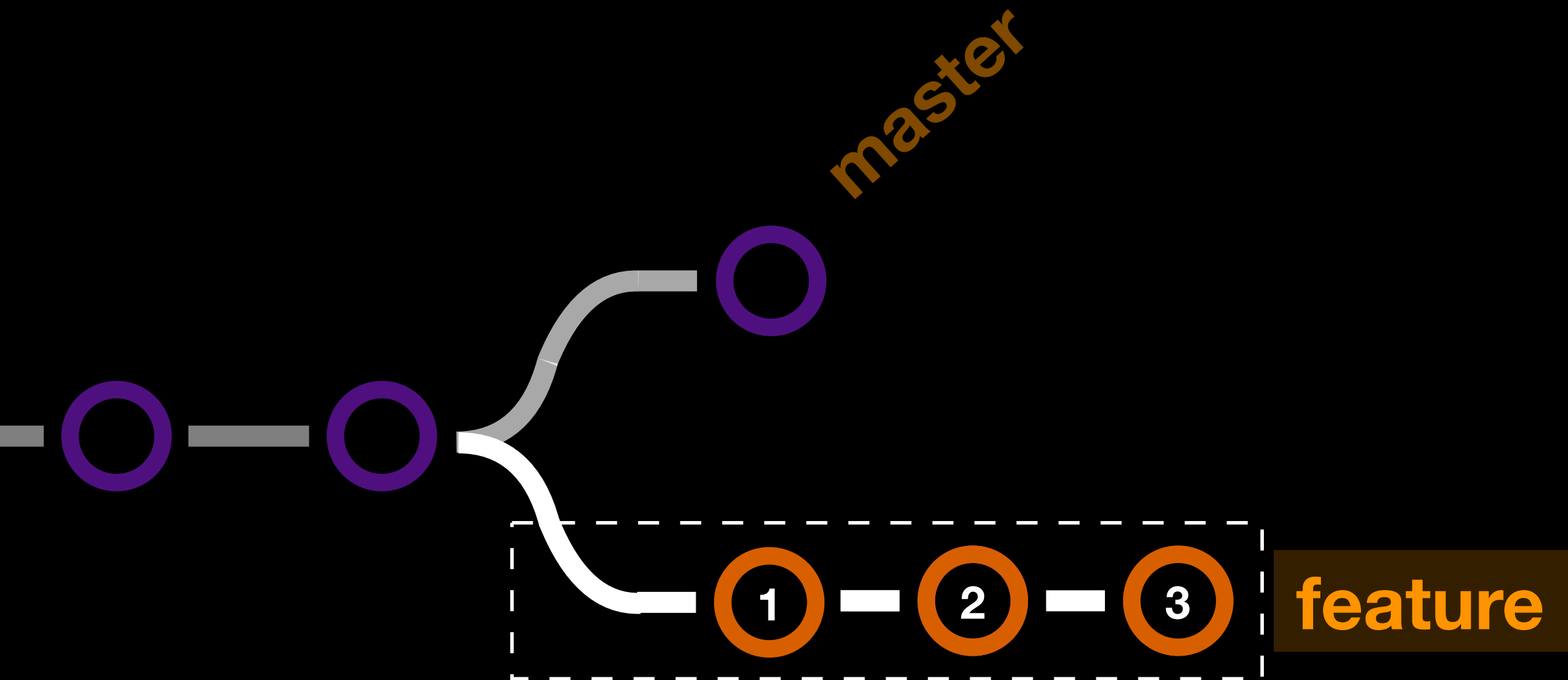
*Give the current branch a new base.
(recreate all the commits of the current
branch on <branch>)*

Example: Continue the development on
<feature> with the new things from <master>

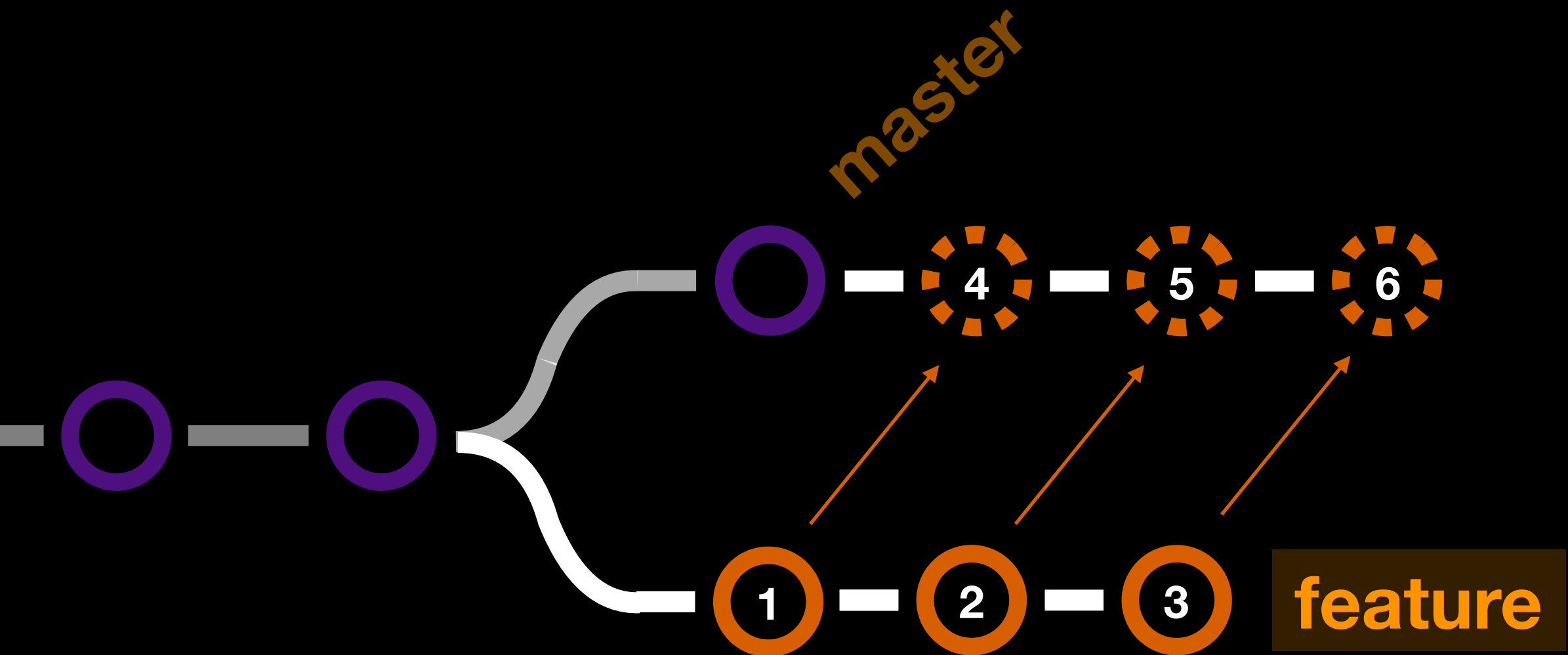
\$ git rebase master



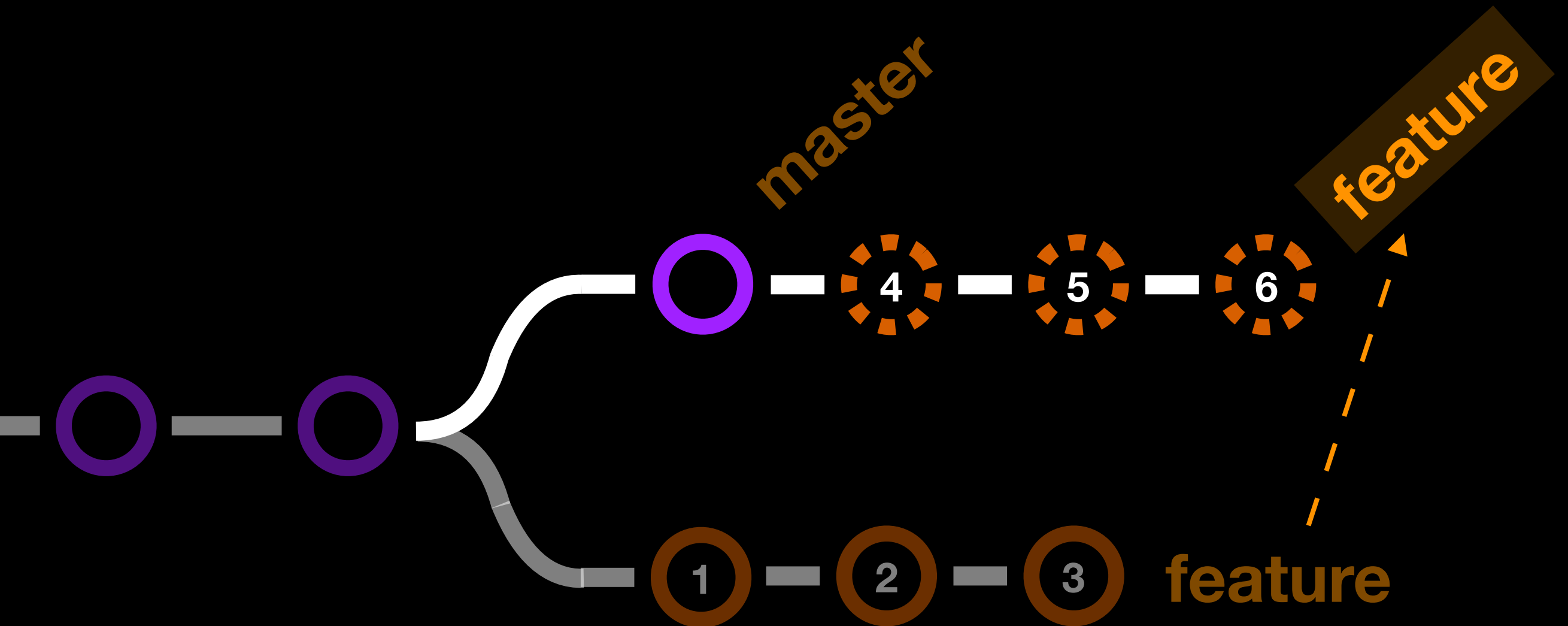
\$ git rebase master



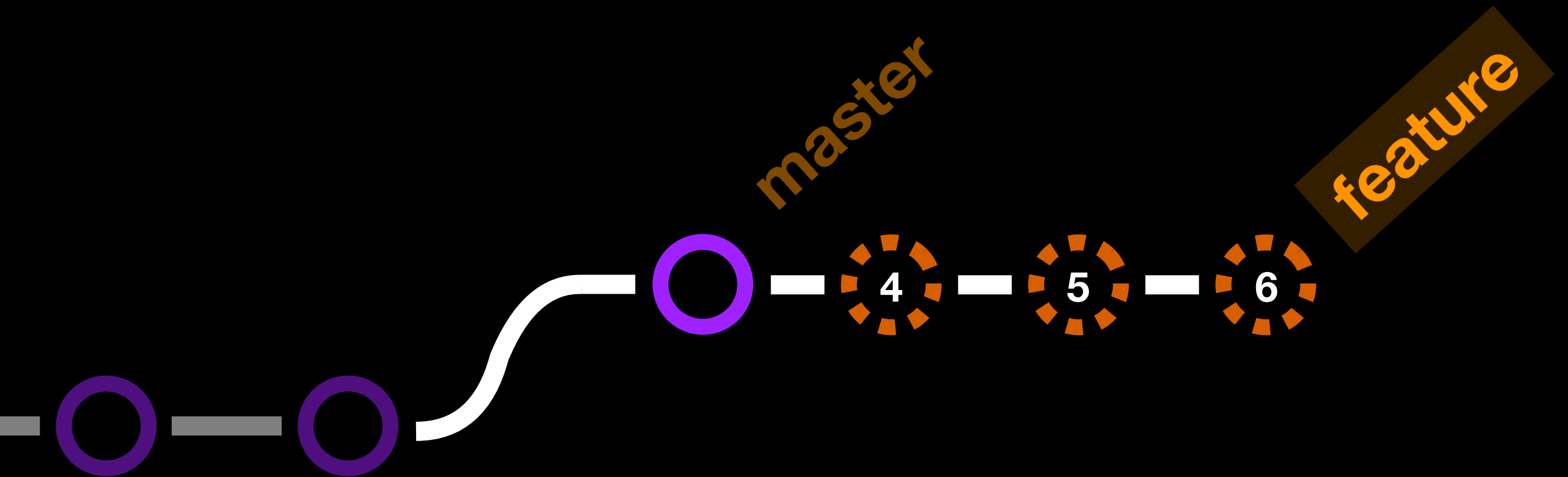
\$ git rebase master



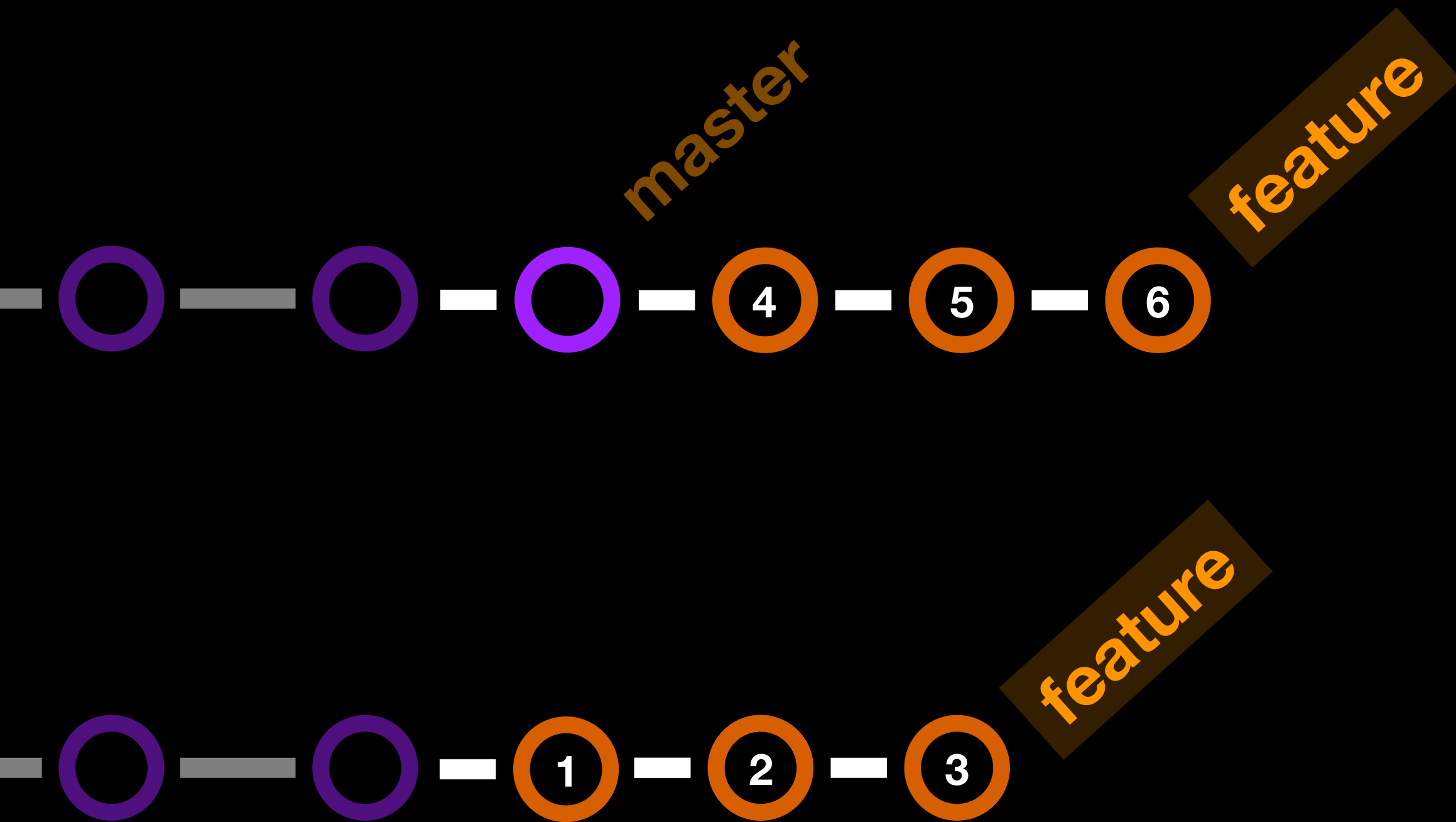
\$ git rebase master



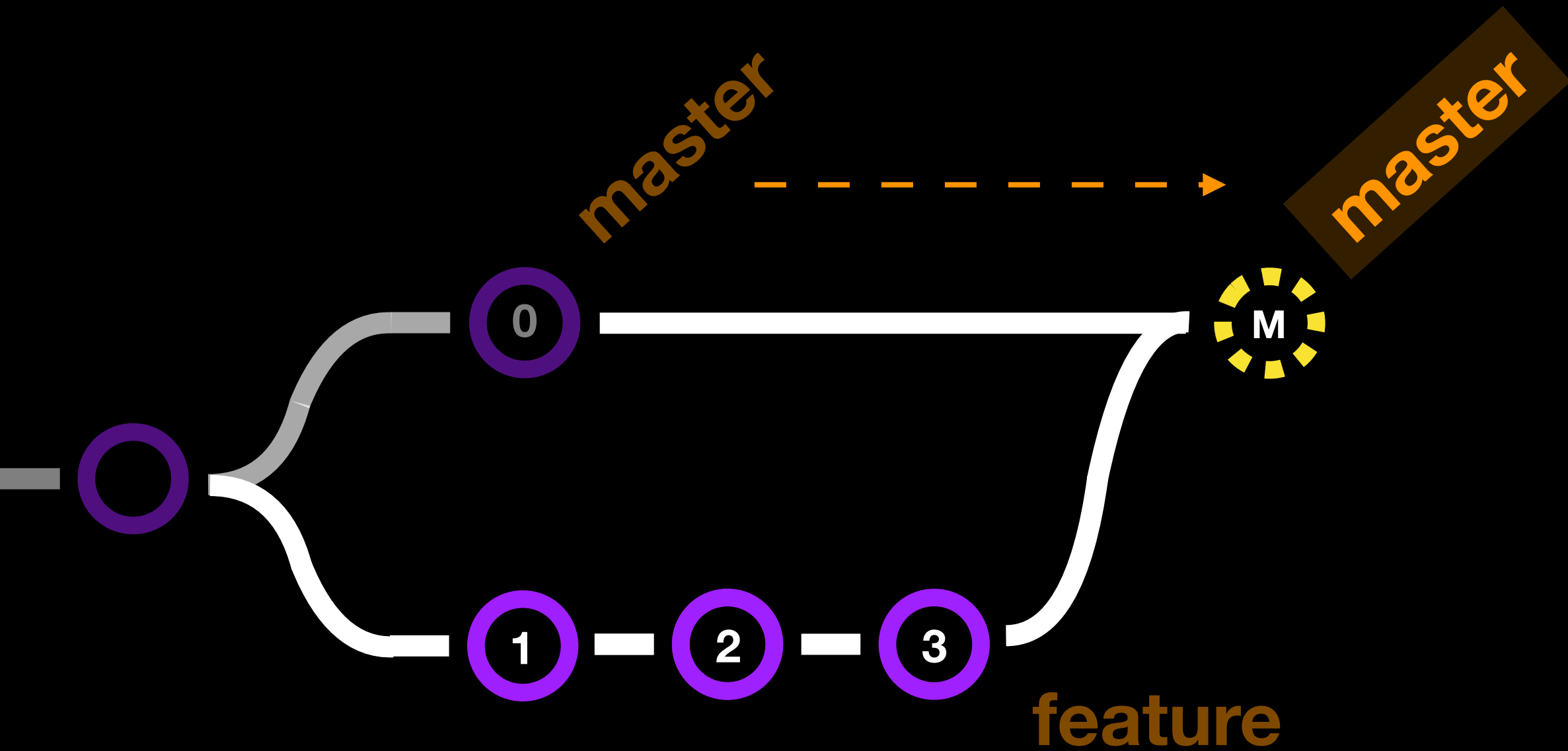
\$ git rebase master



\$ git rebase master



\$ git merge feature





```
git merge <branch>
```

Create a “merge commit” that is the child of both the current branch and <branch>.

Example: Integrate a new feature to the stable branch <master>.





e11baf7fe9b1f

○ 1st Parent: 0000000000...

○ 2nd Parent: 3333333333...

Committed by Nicolas

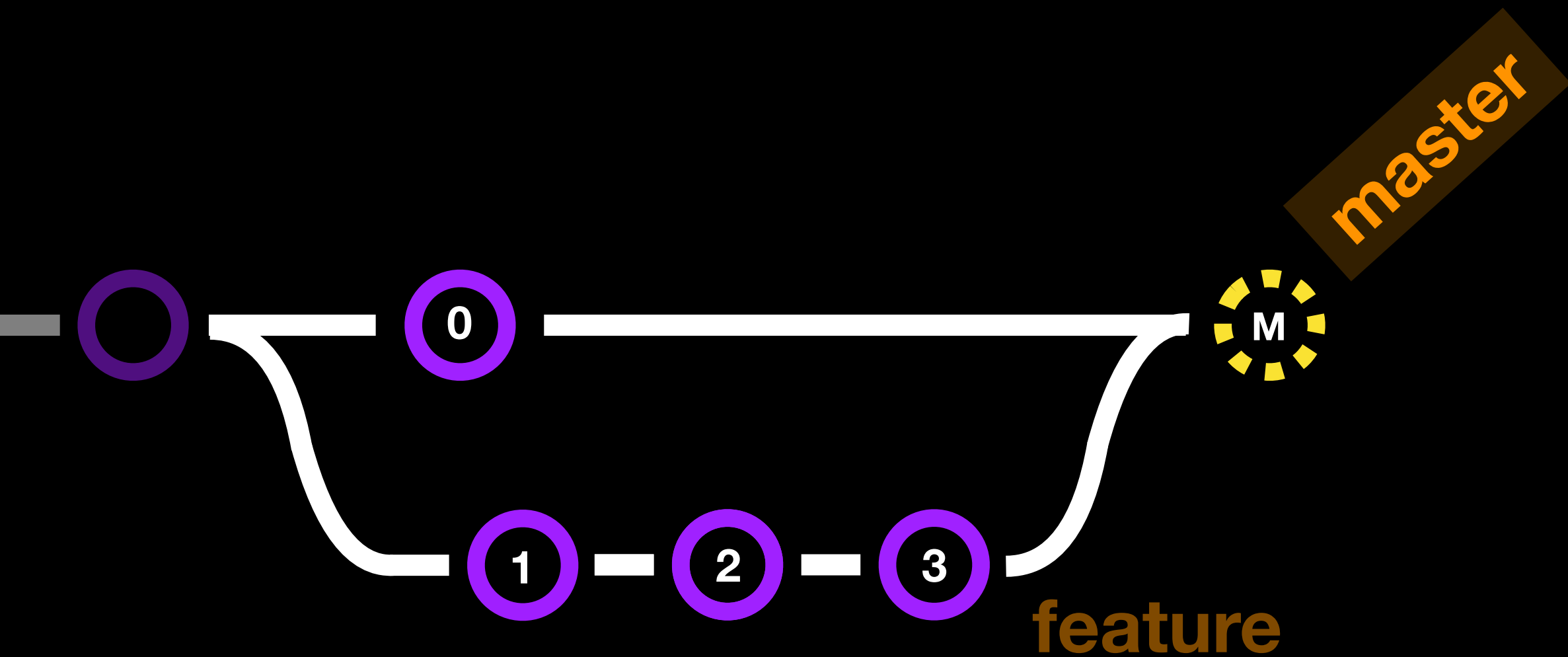
At 16:34 01/03/19

+ Added 1

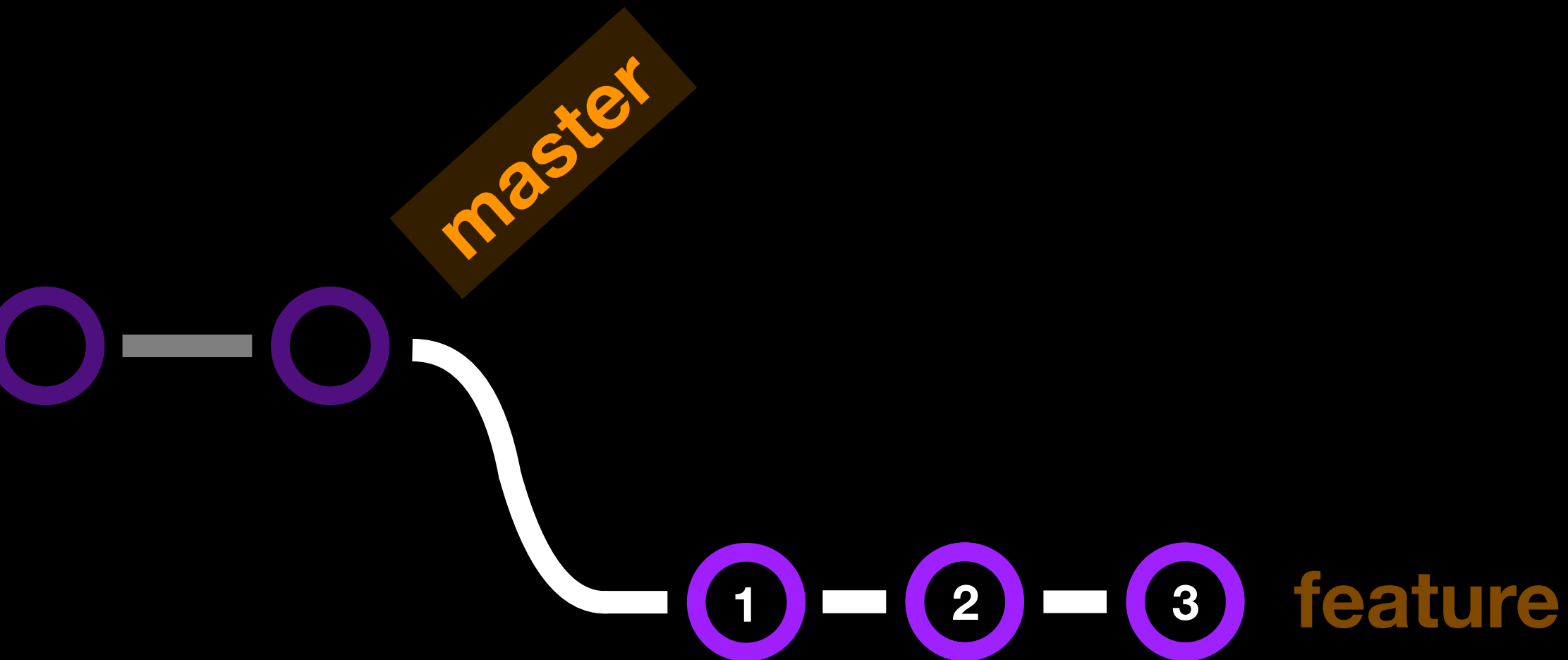
+ Added 2

+ Added 3

\$ git merge feature



\$ git merge feature



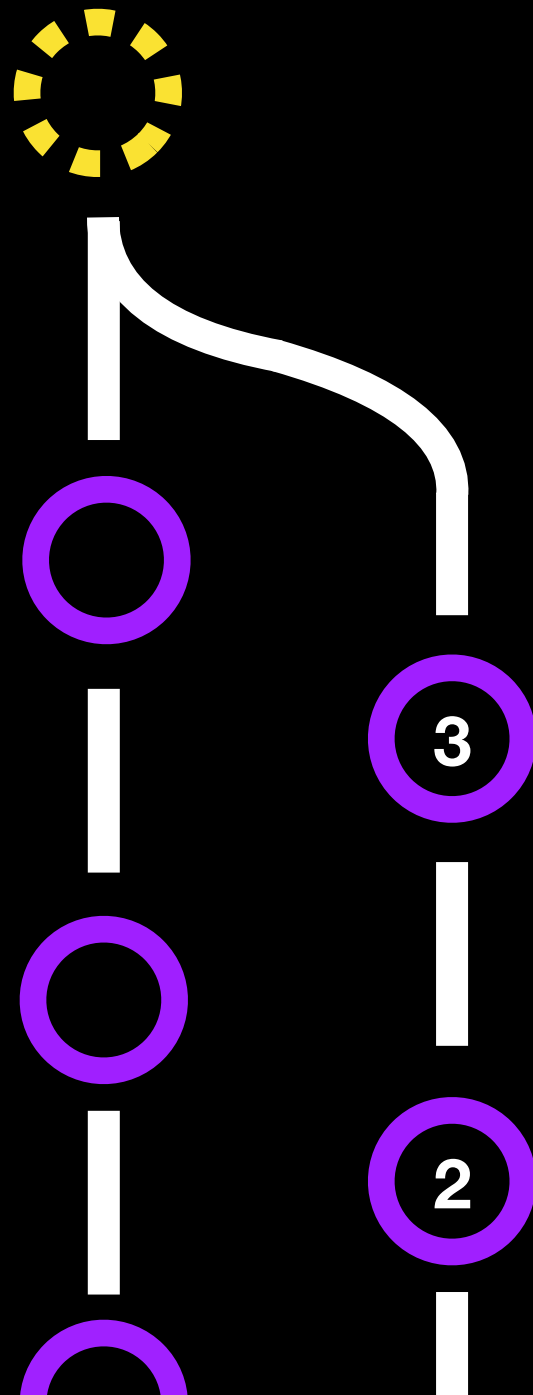
Fast Forward

To force a merge commit

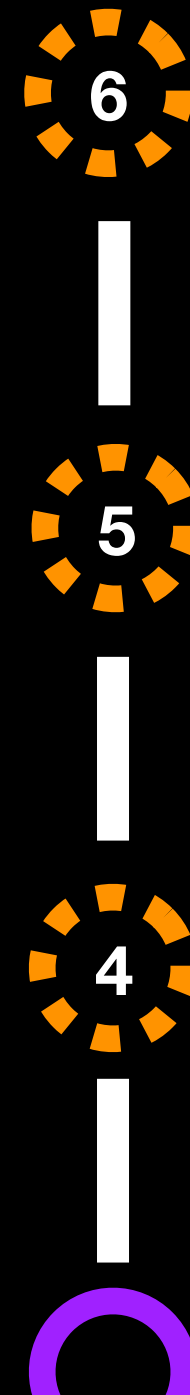
```
git merge --no-ff <branch>
```

Merge vs Rebase

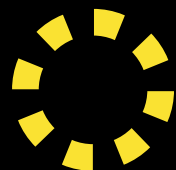
git merge



git rebase



git merge



Preserve the commits

You know better what happened

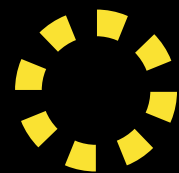
git rebase



Rewrite the commits

You know better what the author
wanted to do

git merge



Preserve the commits

You know better what happened

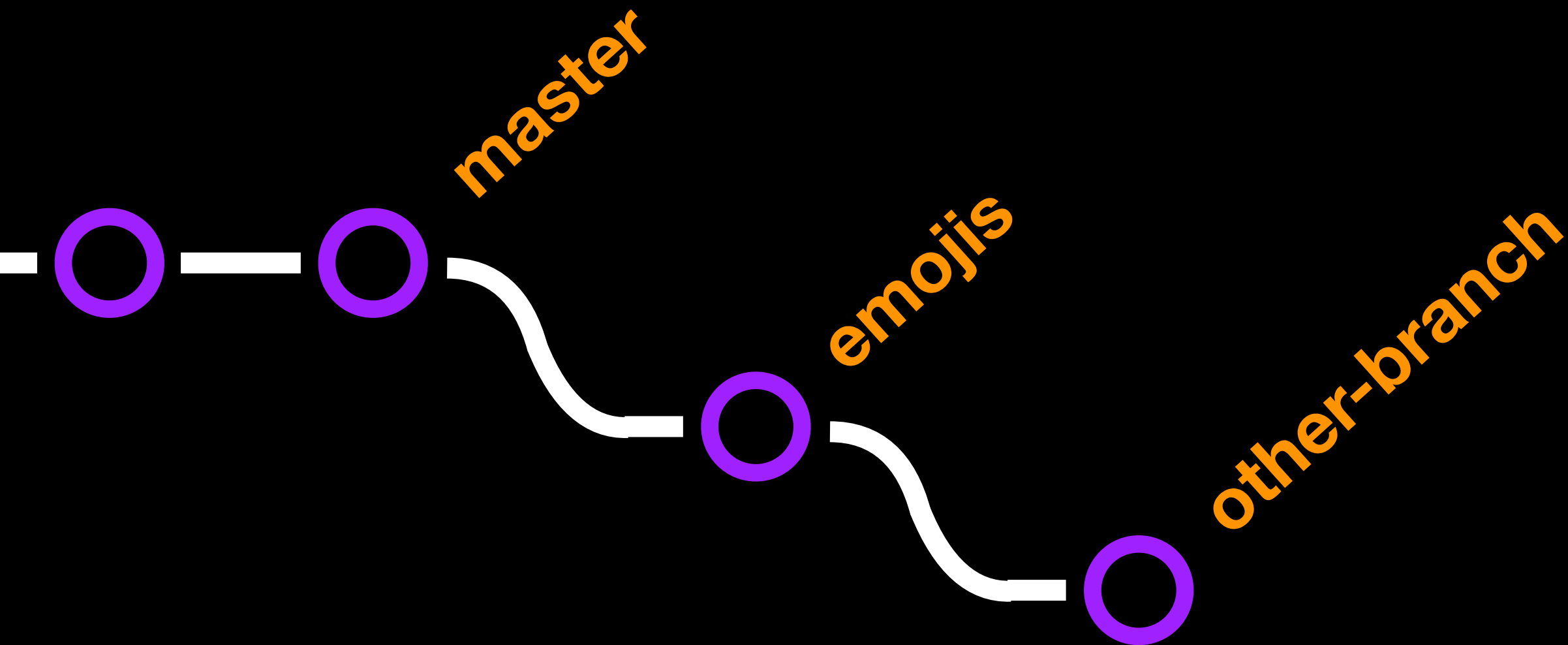
git rebase



Rewrite the commits

**You know better what the author
wanted to do**

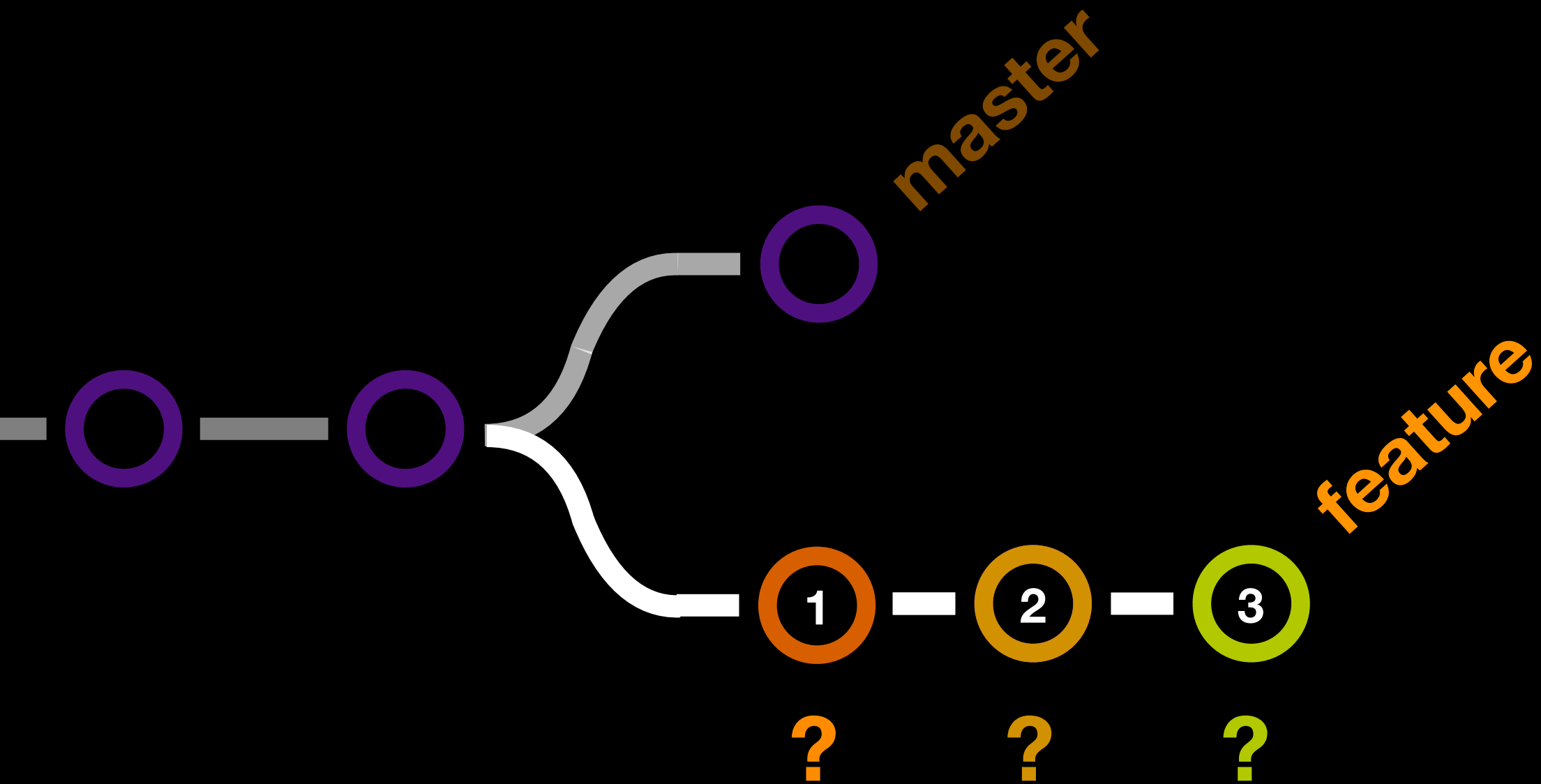
Don't if the commits are used



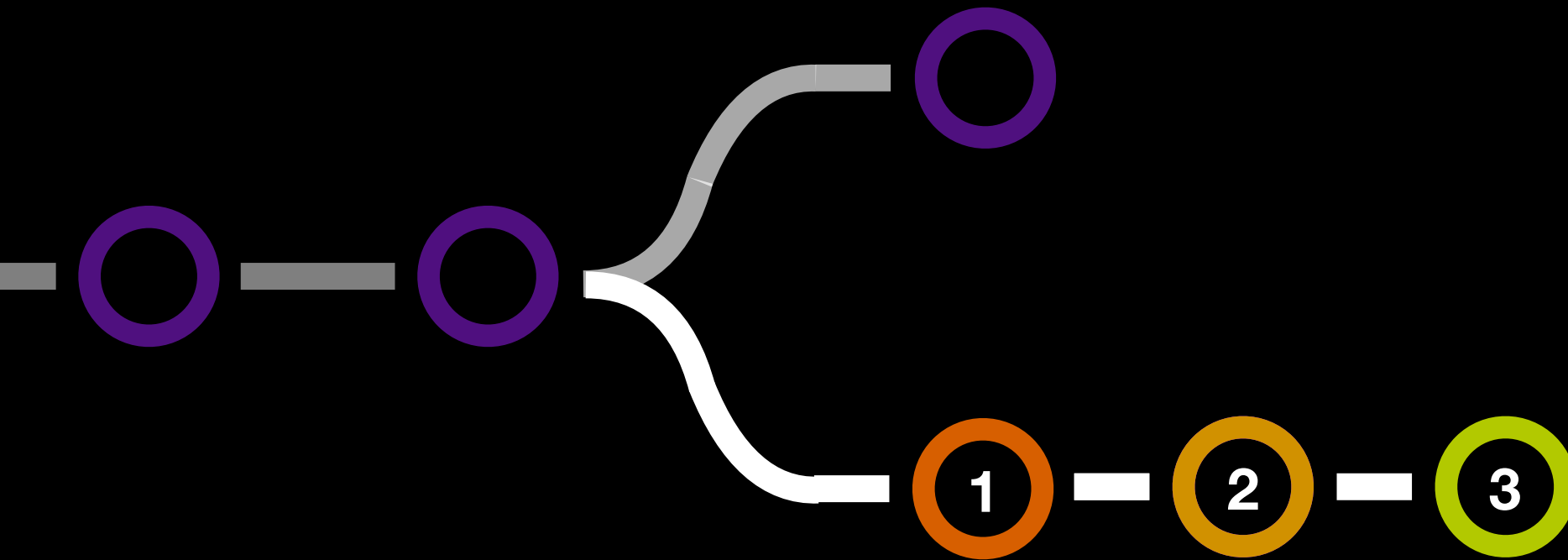
git rebase

git rebase --interactive

\$ git rebase —interactive



\$ git rebase —interactive



1111111111

pick

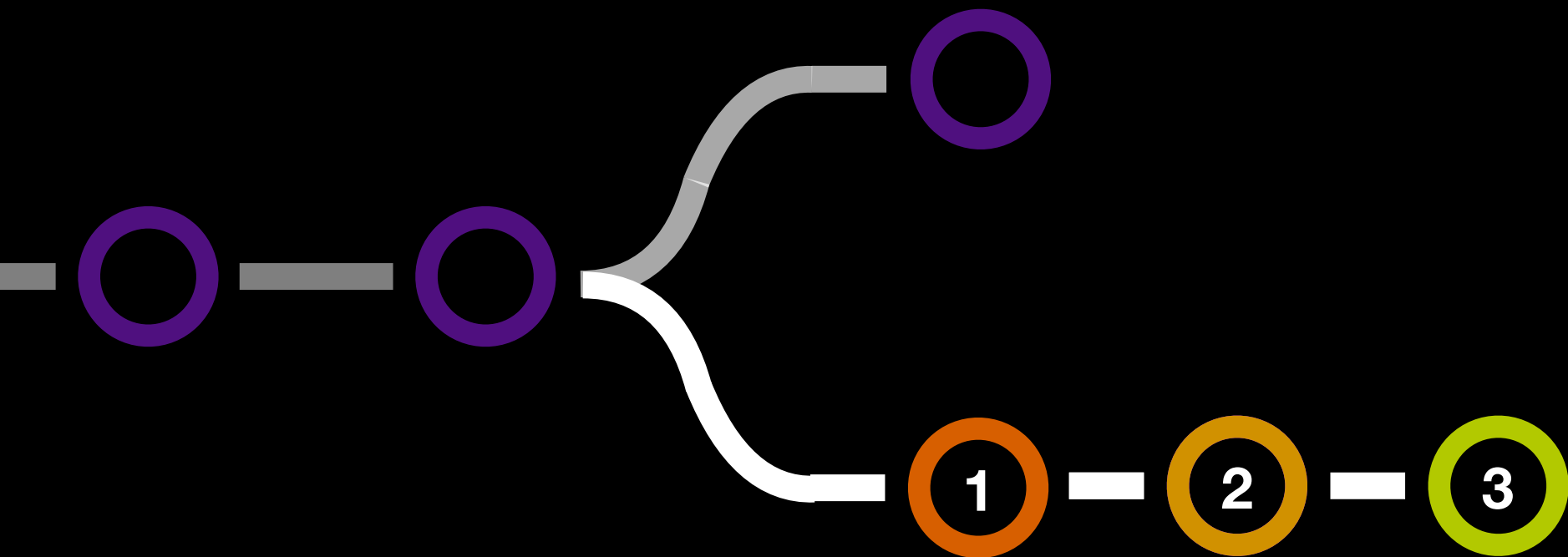
2222222222

pick

3333333333

pick

\$ git rebase —interactive



222222222

pick

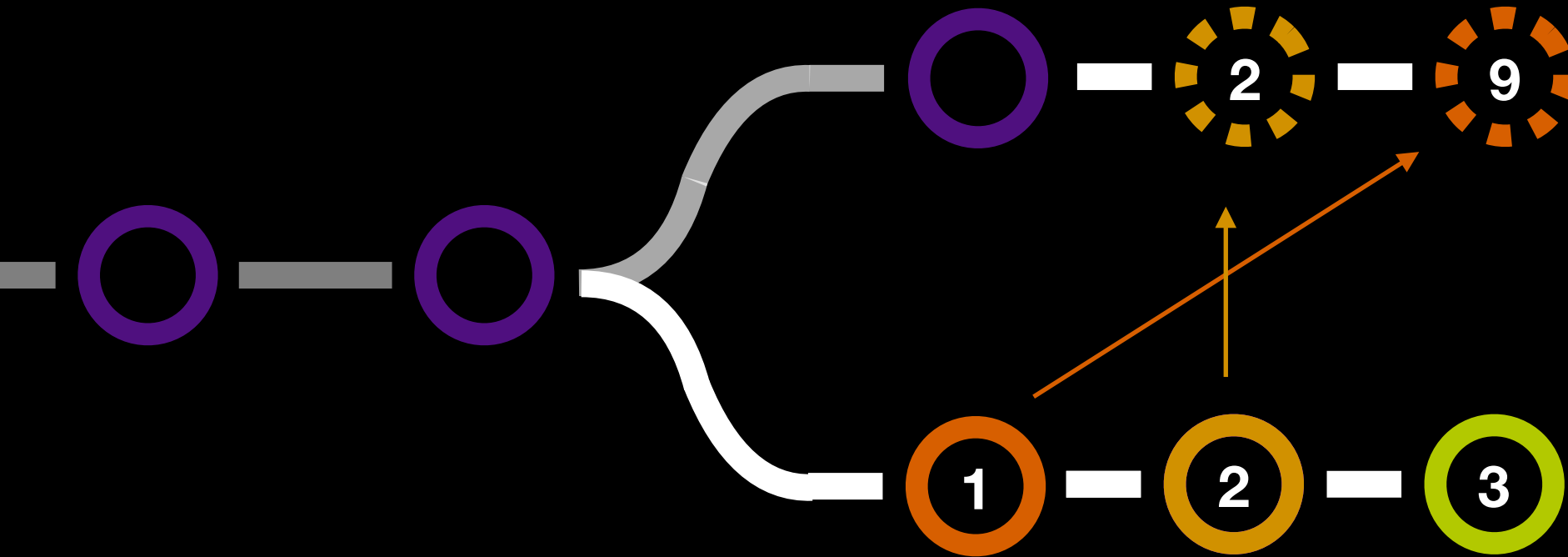
111111111

reword

333333333

drop

\$ git rebase -interactive



222222222

pick

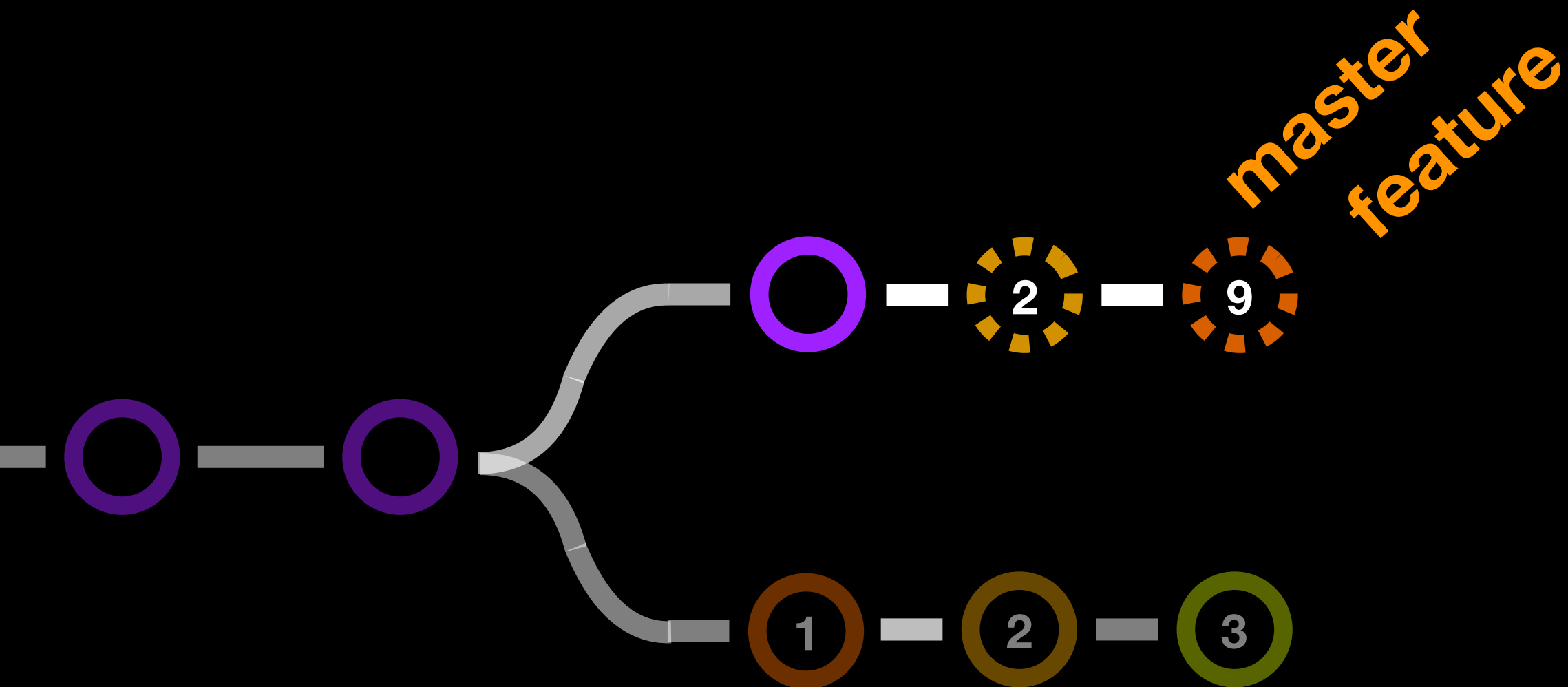
111111111

reword

333333333

drop

\$ git rebase —interactive



pick

Use the commit

reword

Rename

edit

Stop to amend

squash

Merge in the previous commit

drop

Ignore

Please clean up your history

But not if someone is already using it

GitFlow process

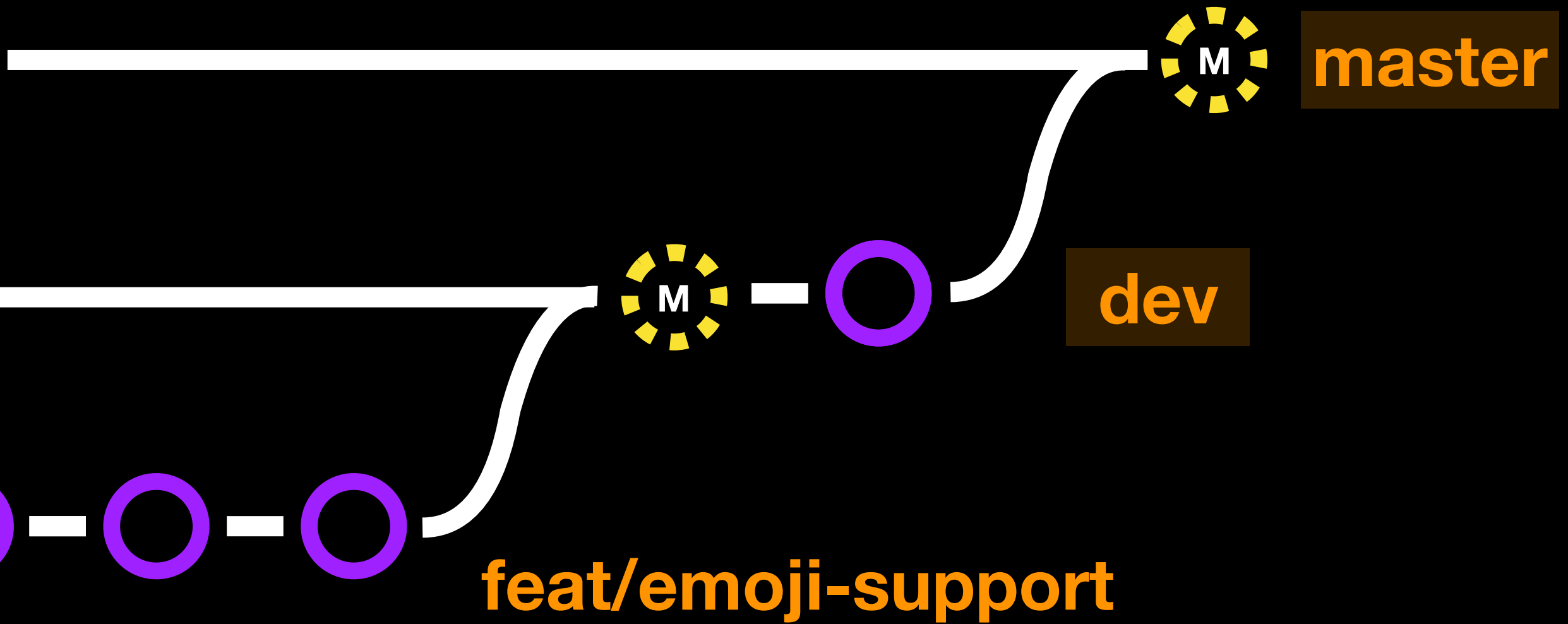


master	Production (stable)
dev	Development (safe)
<branch>	...



feat	Features & improvements
fix	Fixes
docs	Documentation & comments
refactor	Refactor & cleaning
chore	CI, dependencies...
style	Finting, formatting

\$ git merge feature





Commit message

feat: add support for emoji #42

Branch name

feat/add-support-for-emoji-42



En bonus



git reflog

git stash

git pull --rebase