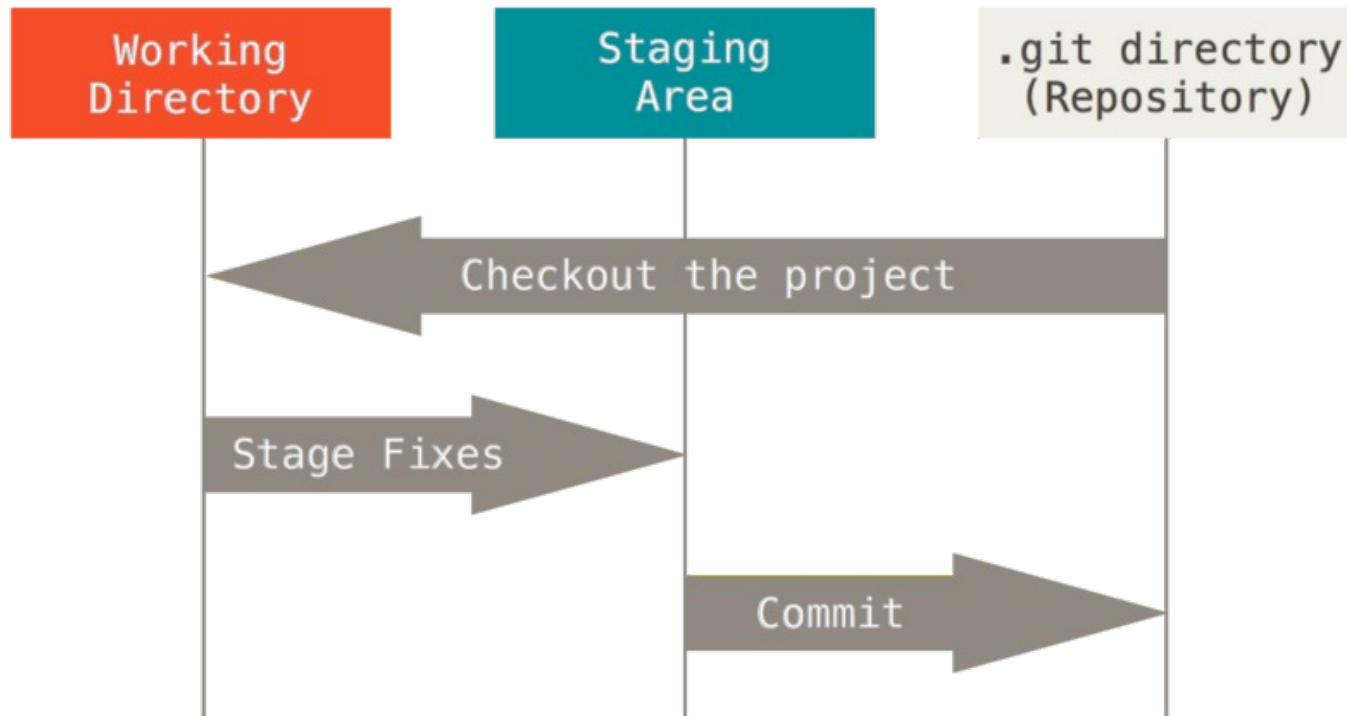


Git

giiit

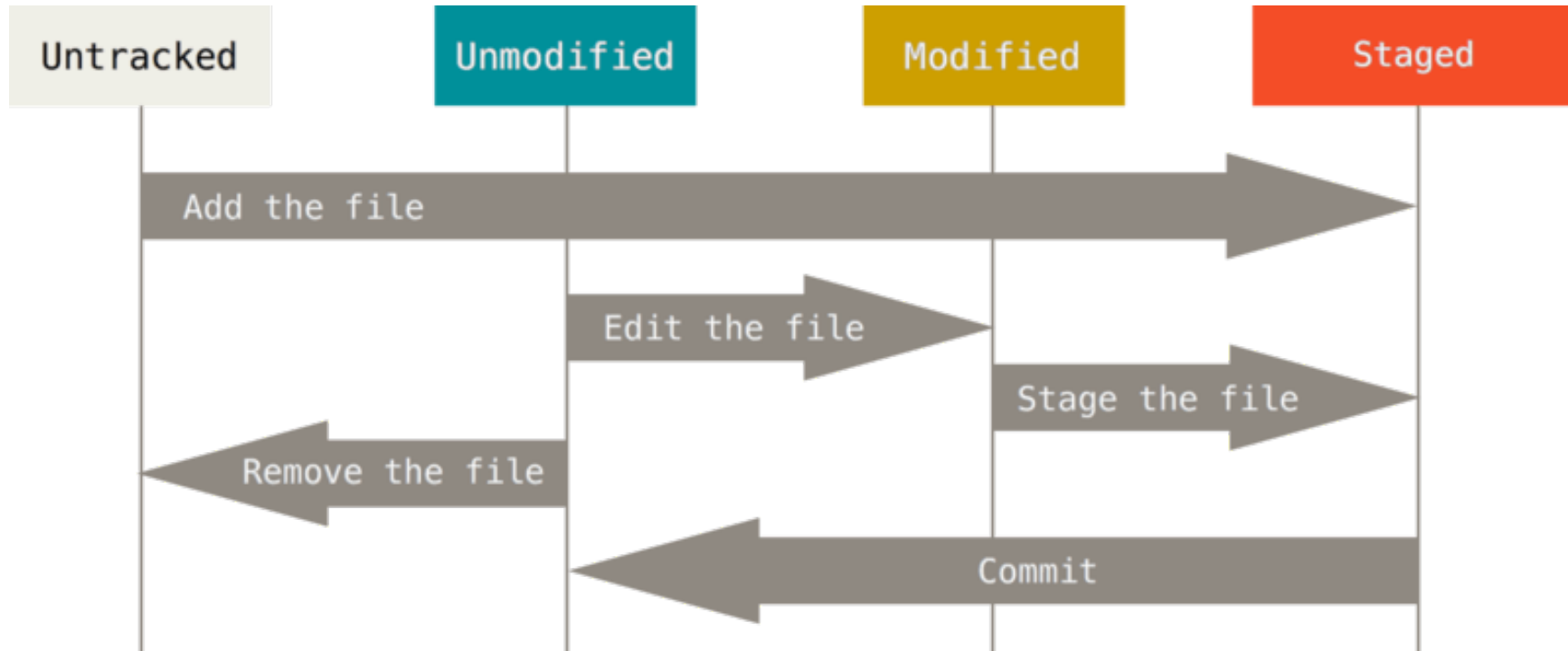
단계



```
>  
> mkdir git_test  
> git init
```

```
drwxr-xr-x eclat staff 320 B Mon Dec 26 22:16:36 2022 ./  
drwxr-xr-x eclat staff 2.7 KB Mon Dec 26 22:17:27 2022 ../  
.rw-r--r-- eclat staff 6.0 KB Tue Dec 6 01:10:15 2022 .DS_Store  
drwxr-xr-x eclat staff 288 B Mon Dec 26 22:16:37 2022 .git/
```

파일 상태



Untracked : 워킹 디렉토리에 있는 파일 중 스냅샷에도 Staging Area에도 포함되지 않은 파일

Tracked : 관리대상 / 이미 스냅샷에 포함되었던 파일

- unmodified
- modified : tracked된 파일이 수정되었을때
- staged : stage에 올려진 상태

```
> echo "happy" > happy.md
> ll
drwxr-xr-x eclat staff 128 B Mon Dec 26 22:26:12 2022 ./
drwxr-xr-x eclat staff 288 B Mon Dec 26 22:25:02 2022 ../
drwxr-xr-x eclat staff 288 B Mon Dec 26 22:25:11 2022 .git/
-rw-r--r-- eclat staff   6 B Mon Dec 26 22:26:12 2022 happy.md
> cat happy.md
happy
> git add happy.md
> git status
현재 브랜치 master

아직 커밋이 없습니다

커밋할 변경 사항 :
(스테이지 해제하려면 "git rm --cached <파일>..."을 사용하십시오)
새 파일 :      happy.md
```

```
> git status
현재 브랜치 master

아직 커밋이 없습니다

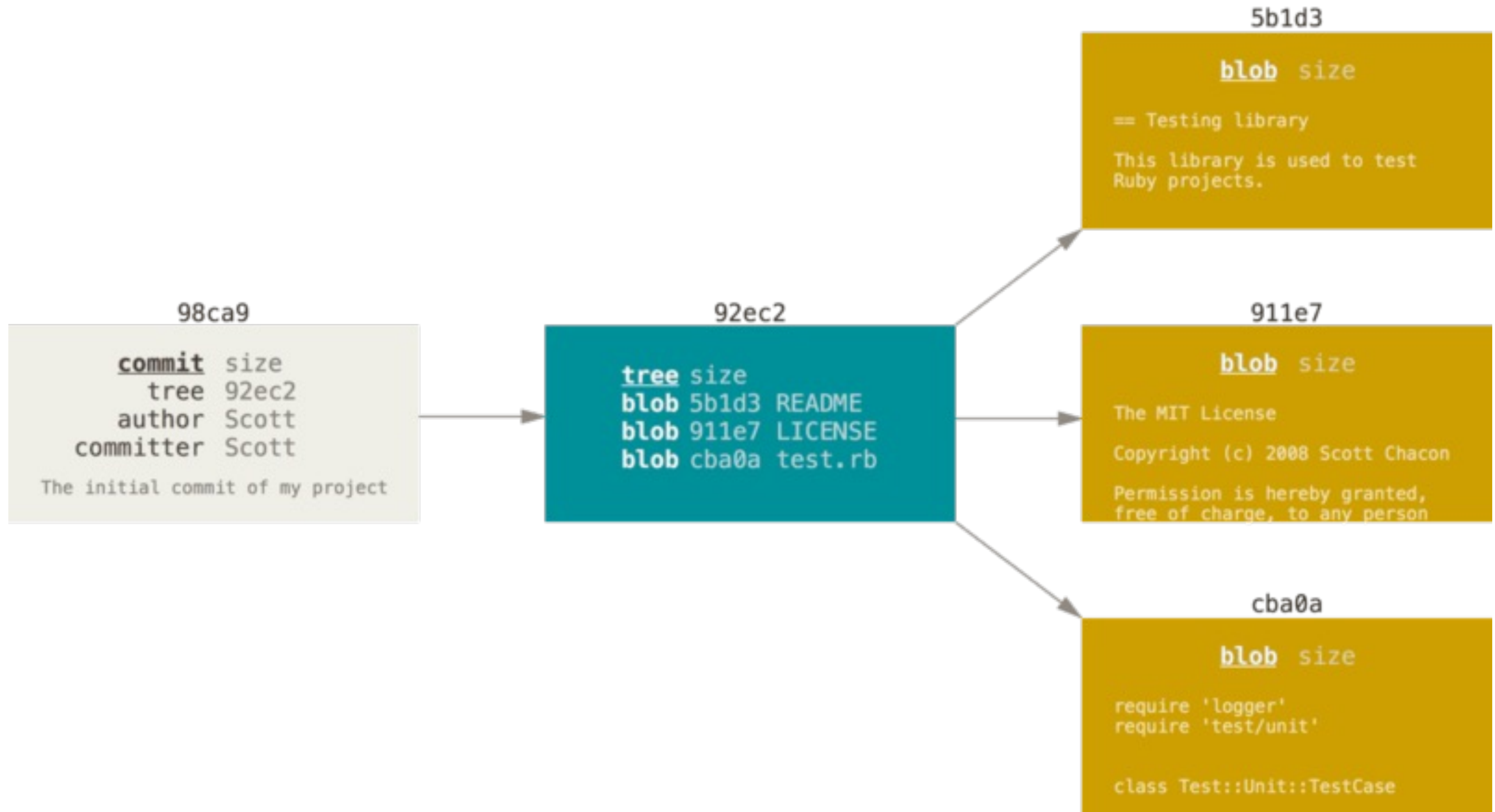
커밋할 변경 사항 :
(스테이지 해제하려면 "git rm --cached <파일>..."을 사용하십시오)
  새 파일 :      happy.md

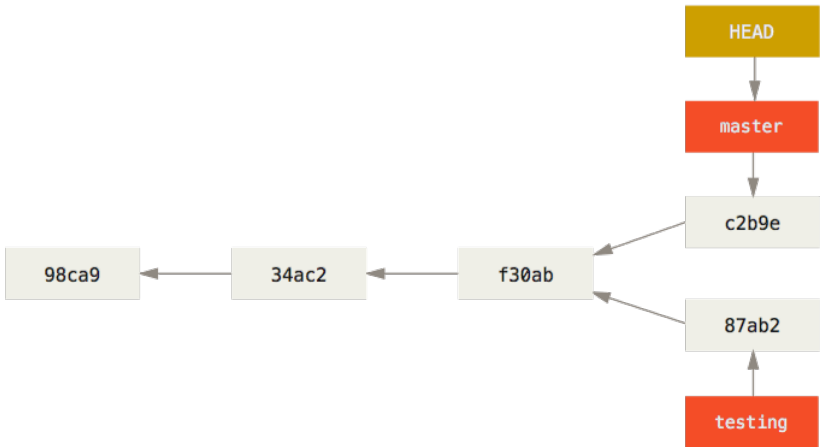
커밋하도록 정하지 않은 변경 사항 :
(무엇을 커밋할지 바꾸려면 "git add <파일>..."을 사용하십시오)
(use "git restore <file>..." to discard changes in working directory)
  수정함 :      happy.md

> cat happy.md
happy new year
```

Staged된 상태에서 수정하면 수정된 파일은 modified 상태가 되고 staged된 상태는 그대로 staged된 상태이다.

커바시 생기는 데이터





```

* 5a3e64b (HEAD -> master) hello 2023
| * 07bcf88 (testing) testing
|/
* 36c153e init
(END)

```

```

> git checkout -b testing
새로 만든 'testing' 브랜치로 전환합니다
> vi happy.md
>
> cat happy.md
happy new year !!!!
> git add happy.md
> git status
현재 브랜치 testing
커밋할 변경 사항:
  (use "git restore --staged <file>..." to unstage)
      수정 함:      happy.md

> git commit -m "testing"
[testing 07bcf88] testing
  1 file changed, 1 insertion(+), 1 deletion(-)
> git log
> git checkout main
error: pathspec 'main' did not match any file(s) known to git
> git checkout master
'master' 브랜치로 전환합니다
> git log --oneline --decorate --graph --all
> vi happy.md
> git add .
> git commit -m "hello 2023"
[master 5a3e64b] hello 2023
  1 file changed, 4 insertions(+)
> git log --oneline --decorate --graph --all
> cat happy.md
happy

hello 2023
~/develop/git_test master
>

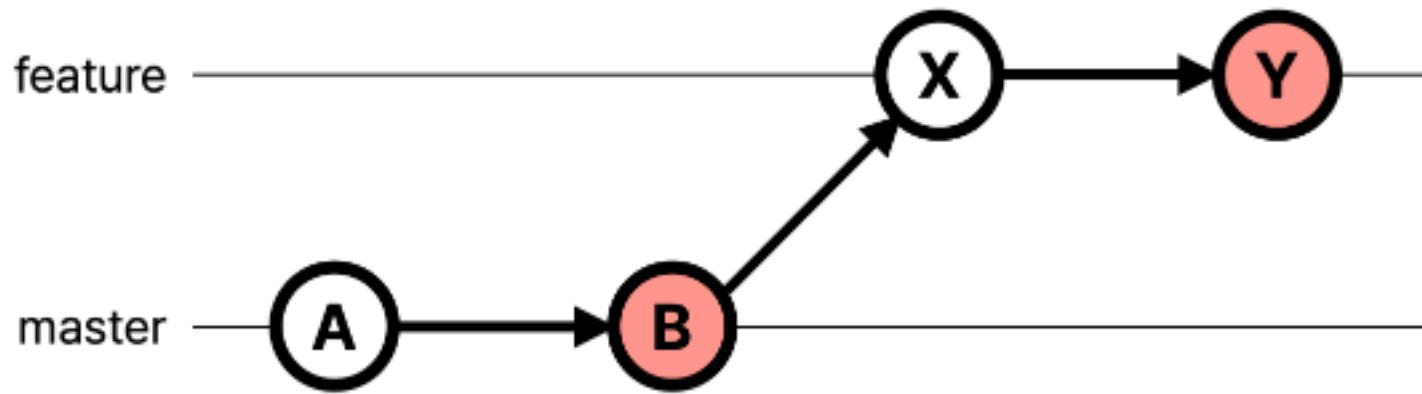
```

Merge

서로 다른 파일을 병합

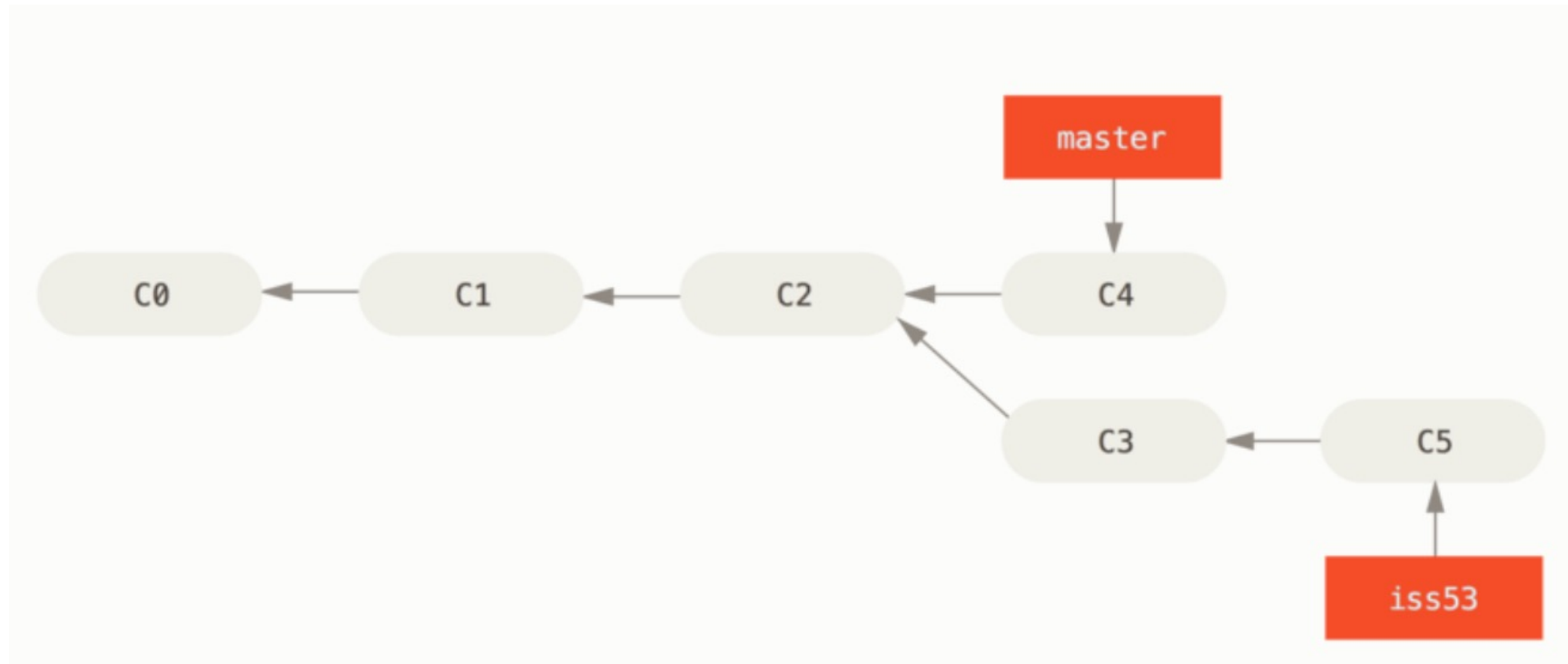
Fast-forward

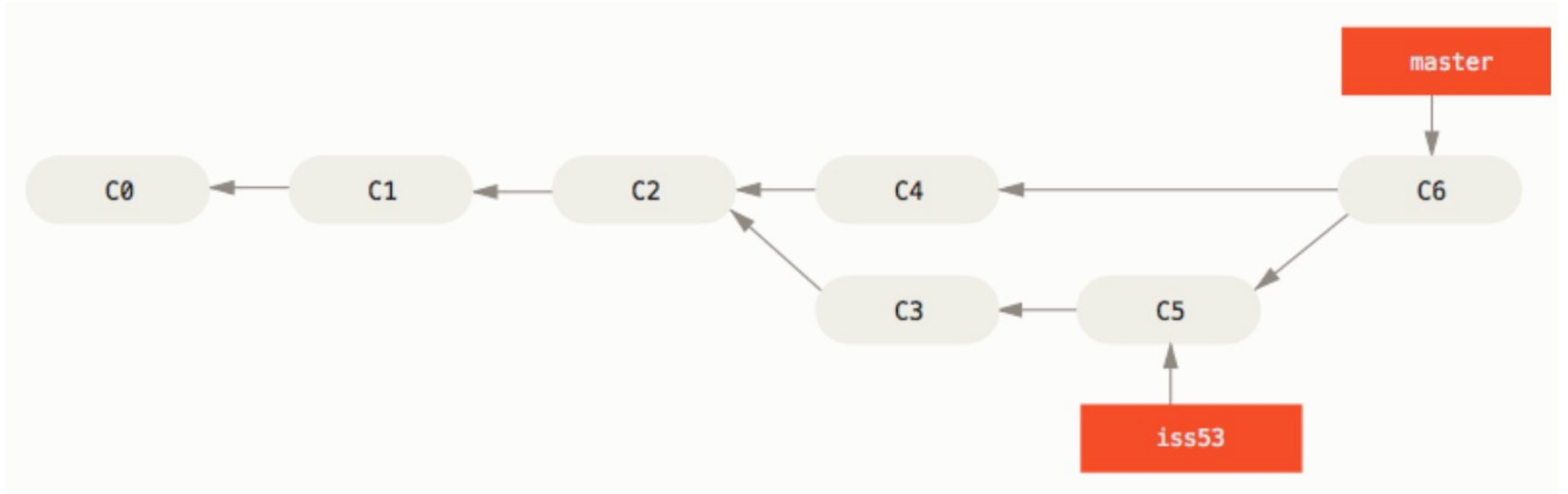
branch 간의 병합을 진행할 때 커밋이 생기지 않고 merge 명령어를 실행하는 브랜치의 Head Commit이 병합되는 branch의 Head Commit으로 이동되는 방식



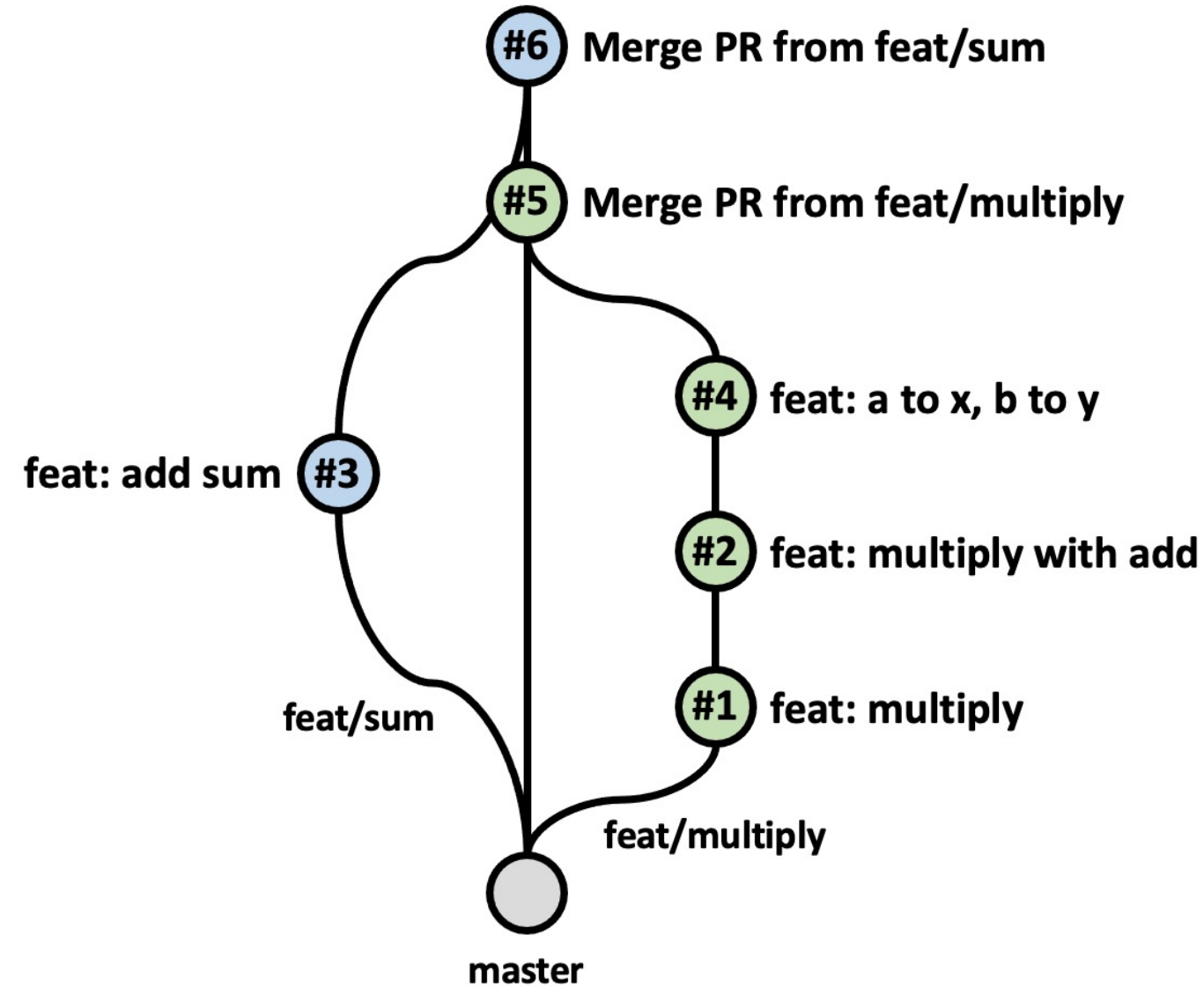
3 way

서로 다른 브랜치에 공통되는 base branch를 기점으로 충돌을 최소화 시키는 방법





Merge commit



Commits on Jun 22, 2021

Merge pull request [#21](#) from inmoonlight/feat/sum ...

inmoonlight committed now

Merge pull request [#22](#) from inmoonlight/feat/multiply ...

inmoonlight committed 14 seconds ago

fix: a to x, b to y

inmoonlight committed 2 hours ago

feat: add sum

inmoonlight committed 2 hours ago

feat: multiply with add

inmoonlight committed 4 hours ago

feat: multiply

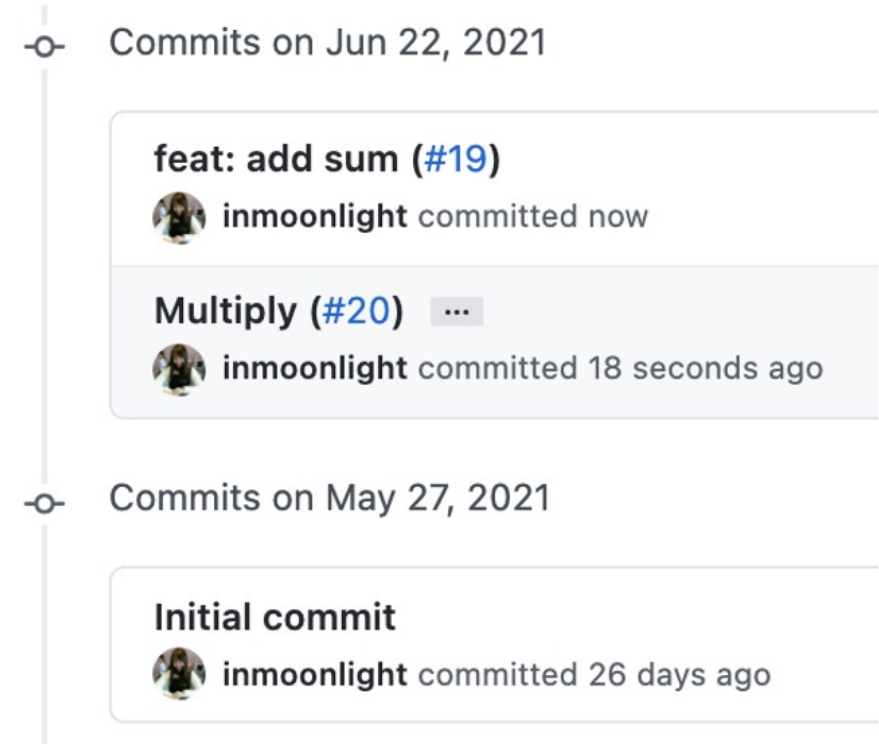
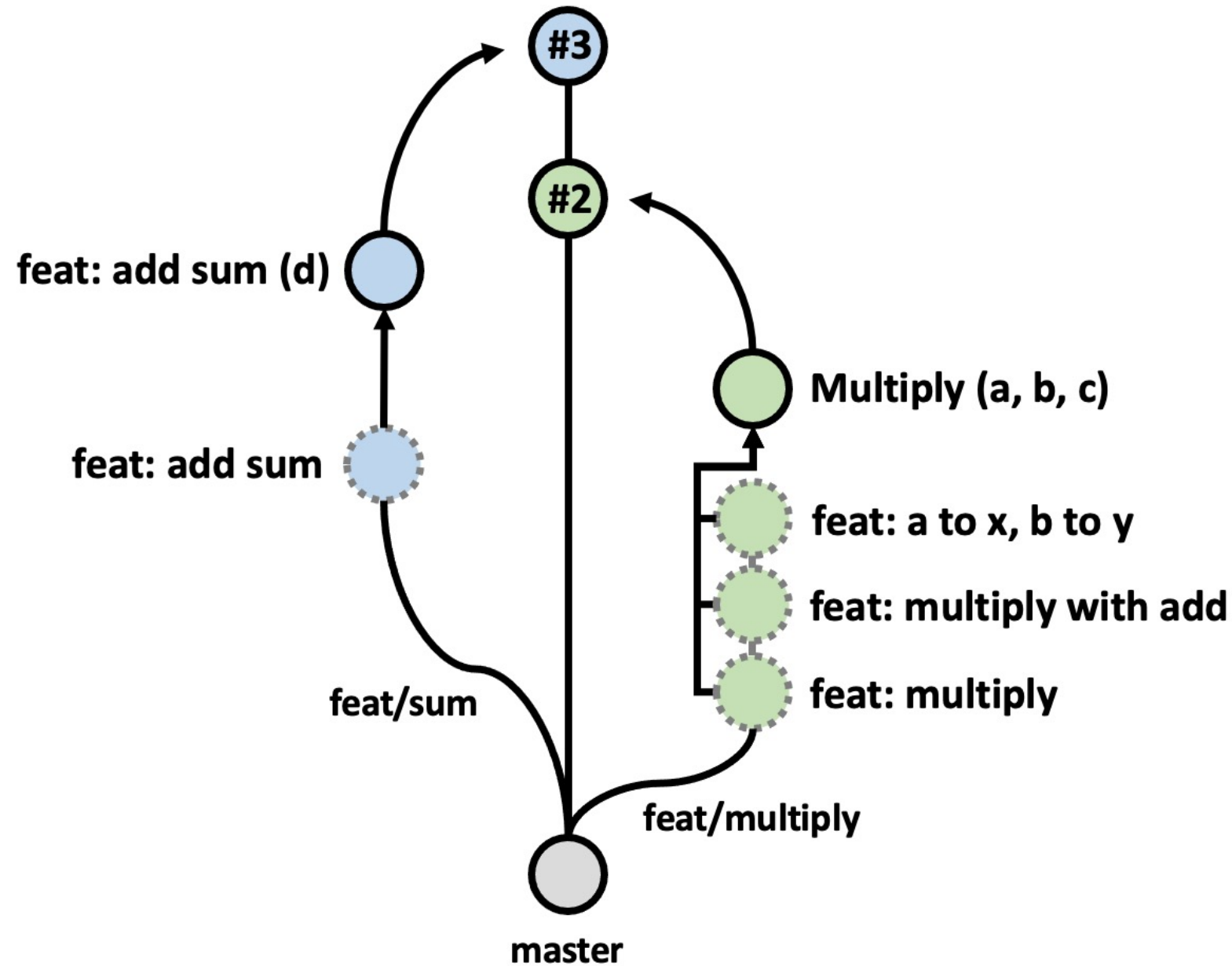
inmoonlight committed 5 hours ago

Commits on May 27, 2021

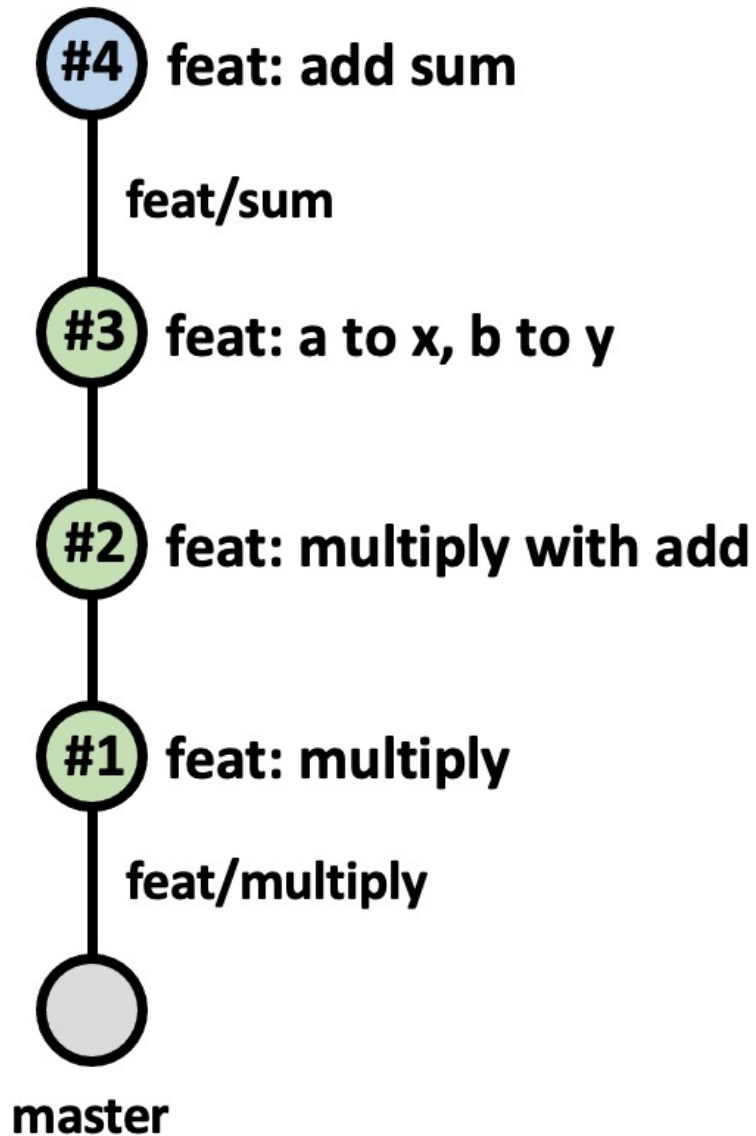
Initial commit

inmoonlight committed 26 days ago

Squash and merge



Rebase and merge



Commits on Jun 22, 2021

feat: add sum



inmoonlight committed 43 minutes ago

fix: a to x, b to y



inmoonlight committed 43 minutes ago

feat: multiply with add



inmoonlight committed 43 minutes ago

feat: multiply



inmoonlight committed 43 minutes ago



Commits on May 27, 2021

Initial commit



inmoonlight committed 26 days ago