

25~27

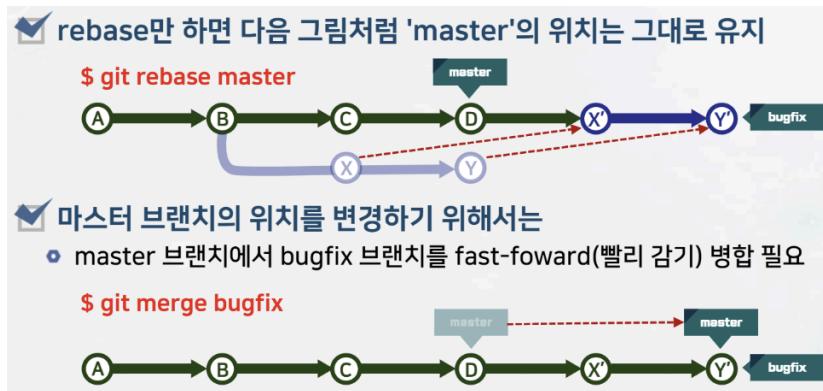
base

현재 두 브랜치의 공통 조상

rebase

브랜치에서 base를 바꾸는 재배치

```
$ git rebase master
```



rebase 충돌

이동되는 X와 Y의 내용이 'master'의 C,D 커밋들과 충돌하는 부분이 생길 수 있음

해결 절차 :

- 파일 수정
- 파일 추가 : \$ git add <수정파일>
- rebase 계속 수행, 마지막 메시지 수정 : \$ git rebase - -continue

(이해를 돋기 위해 chatGPT 사용)

1단계: 충돌난 파일을 열어 직접 수정

충돌이 나면 Git이 멈추고 다음과 같은 메시지를 보여준다:

CONFLICT (content): Merge conflict in <파일명>

해당 파일을 열면 다음 형식으로 표시됨:

```
<<<<<< HEAD(or current)
```

현재 적용 중인 커밋 내용

```
=====
```

다음에 적용하려는 커밋 내용

```
>>>>> <branch>
```

👉 해야 할 일

- 두 코드 중 하나를 선택하거나
- 둘을 합쳐서 새로운 코드로 만들고
- **충돌 표시(<<<<<<, =====, >>>>>)** 모두 삭제

■ 2단계: 수정한 파일 stage 올리기

수정 후:

```
git add <충돌해결한 파일>
```

파일이 여러 개면:

```
git add .
```

■ 3단계: Rebase 계속 진행

충돌을 해결하면 다음 명령으로 이어서 진행:

```
git rebase --continue
```

이제 Git은 다음 커밋을 다시 적용하려 한다.

그리고 다음 커밋에서도 충돌이 있다면 다시 충돌 처리 반복.

■ 4단계: 모든 충돌을 해결하면 rebase가 완료됨

모든 커밋이 정상 적용되면 Git은 다음 메시지를 출력:

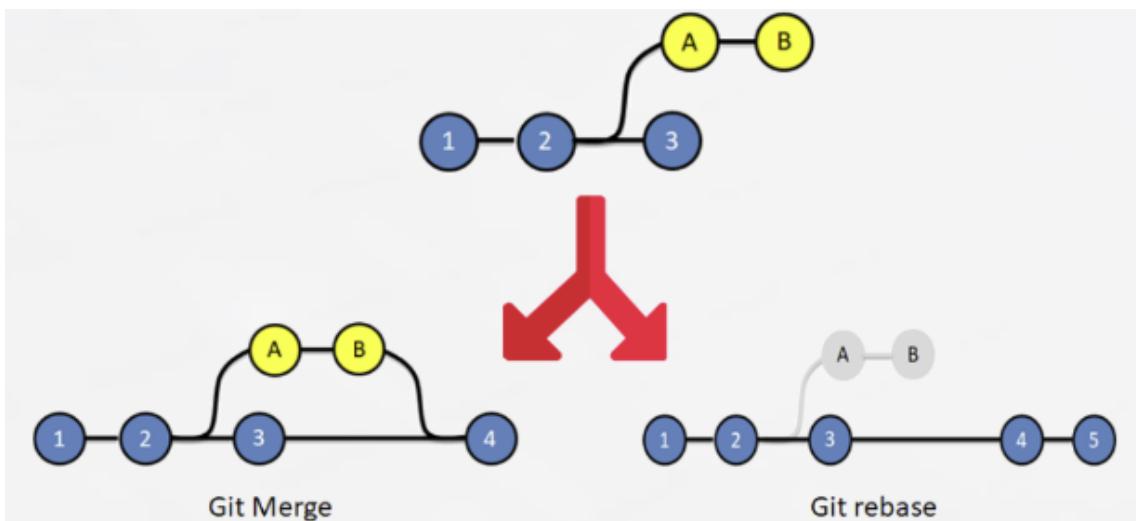
Successfully rebased and updated refs/heads/<branch>

✓ 끝!

3-way의 merge와 rebase

merge : 변경 내용의 이력이 모두 그대로 남아있음

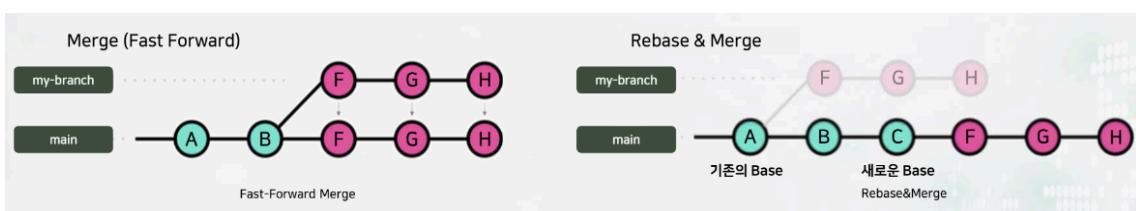
rebase : 원래의 커밋 이력이 변경됨. 좀 더 깨끗한 이력(선행)



fast-forward의 merge와 rebase

merge : 조상에 위치한 브랜치에서 선행 브랜치의 마지막으로 이동

rebase : 병합할 브랜치 마지막 커밋을 새로운 베이스로 수정



rebase 방법

- 일반적 rebase 방법

```
$ git checkout topic
```

```
$ git rebase main
```

```
$ git checkout main
```

```
$ git merge topic
```

- 다른 rebase 방법 : 어느 브랜치든 main topic 순서로 재배치 방법

```
$ git rebase main topic
```

```
$ git checkout main
```

```
$ git merge topic
```

커밋 옵션 - --amend

- 설정된 편집기로 최신 커밋 메시지 수정 : \$ git commit --amend
- 설정된 편집기로 최근 커밋 메시지를 직접 입력해 수정 : \$ git commit --amend -m 'new message'
- 메시지 수정 없이 최신 커밋으로 수정 : \$ git commit --amend --no-edit

이전 커밋 HEAD~2..HEAD까지 각각의 커밋을 수정

- \$ git rebase --interactive HEAD~3
- \$ git rebase -i HEAD~3

주요 rebase -i 대화형 명령어

- p(ick) : 해당 커밋을 수정하지 않고 그대로 사용
- r(eword) : 개별 커밋 메시지를 다시 작성
- s(quash) : 계속된 이후 커밋을 이전 커밋에 결합(코드 변경사항은 합쳐짐 / 커밋 메시지는 합쳐서 하나로 선택하거나 수정함)

✓ 예시(chatGPT)

```
pick 1a2b3c 로그인 UI 추가  
squash 4d5e6f 로그인 UI 오타 수정  
squash 7g8h9i 로그인 버튼 색 변경
```

→ 결과:

"로그인 UI 하나의 커밋" 으로 합쳐짐

```
로그인 UI 관련 작업 (오타 수정 + 버튼 색 포함)
```

→ 이력이 깔끔해짐

- d(rop) : 커밋 자체를 삭제

VScode 로 활용