

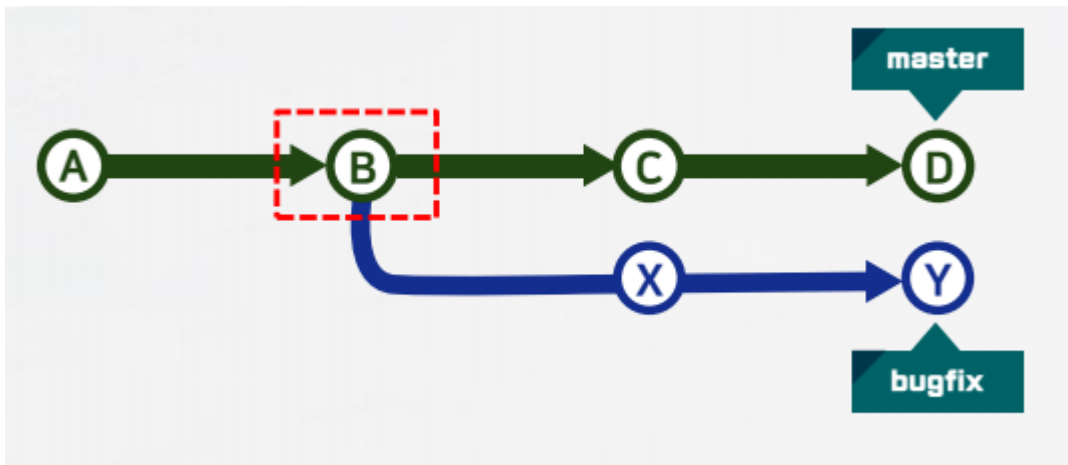


12주차

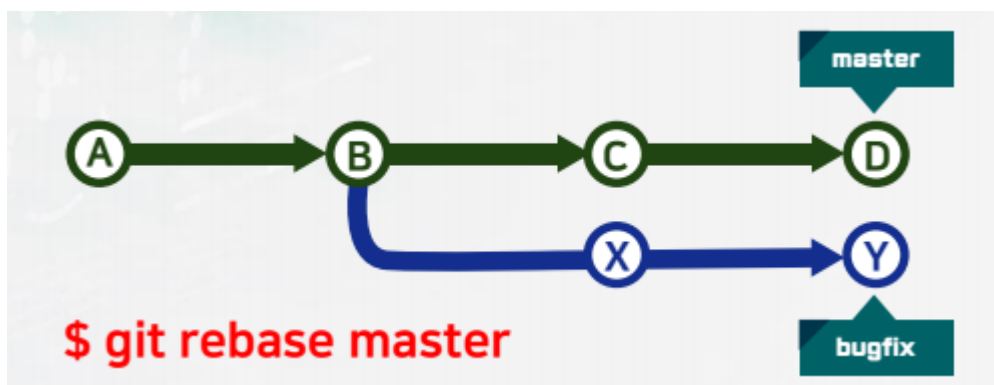
브랜치 리베이스 rebase

병합 rebase

- 3-way 상태에서 base의 이해
 - master 브랜치 커밋 b에서 분기되는 bugfix 브랜치



- 브랜치 bugfix에서 base를 바꾸는 재배치하기 이전



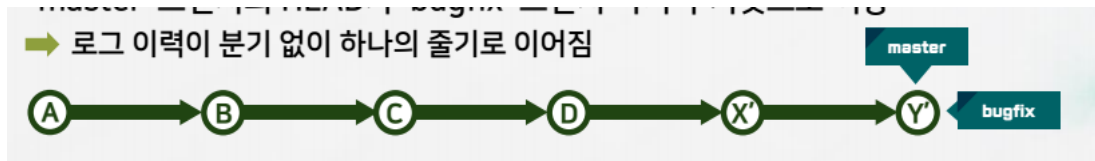
- 재배포 rebase 병합 수행

- base를 수정

- B에서 마스터의 최신 커밋인 D로 수정

- D이후 bugfix를 배치

- 이후 다시 fast-forward 병합 수행 : 이 병합을 직접 다시 해야한다



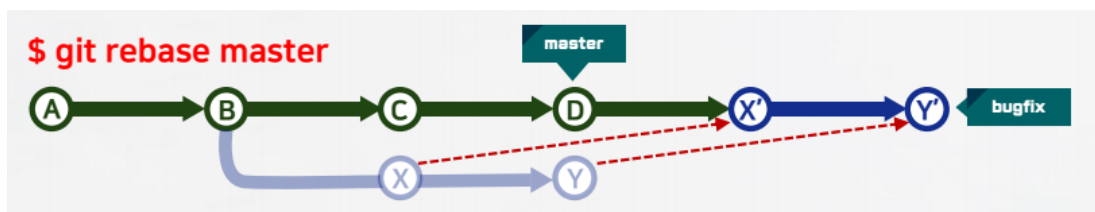
- rebase를 이용한 브랜치 병합 과정

- fast-forward 병합 방식

- master 브랜치 뒤로 bugfix 브랜치의 이력이 이동

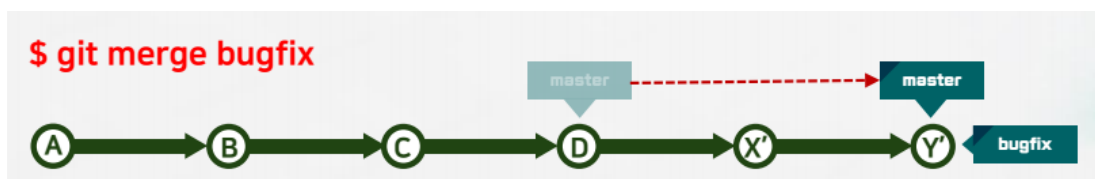
- 충돌 발생 가능

- rebase만 하면 다음 그림처럼 master의 위치 그대로 유지



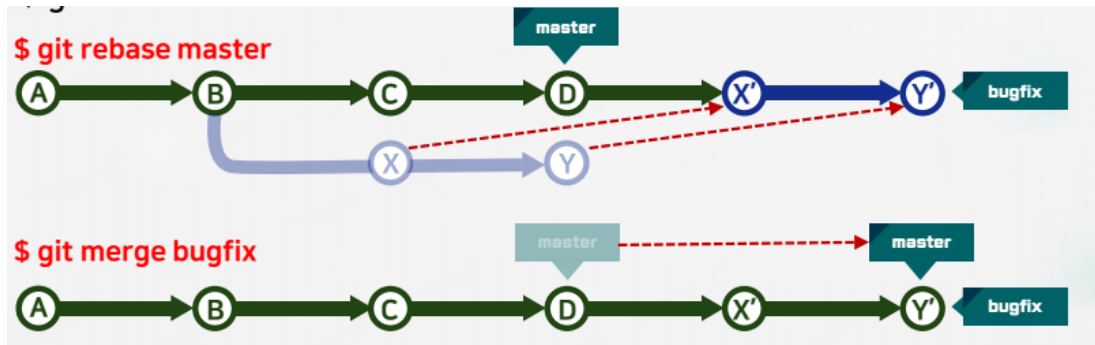
- 마스터 브랜치의 위치를 변경하기 위해서는

- master 브랜치에서 bugfix 브랜치를 fast-foward 병합 필요

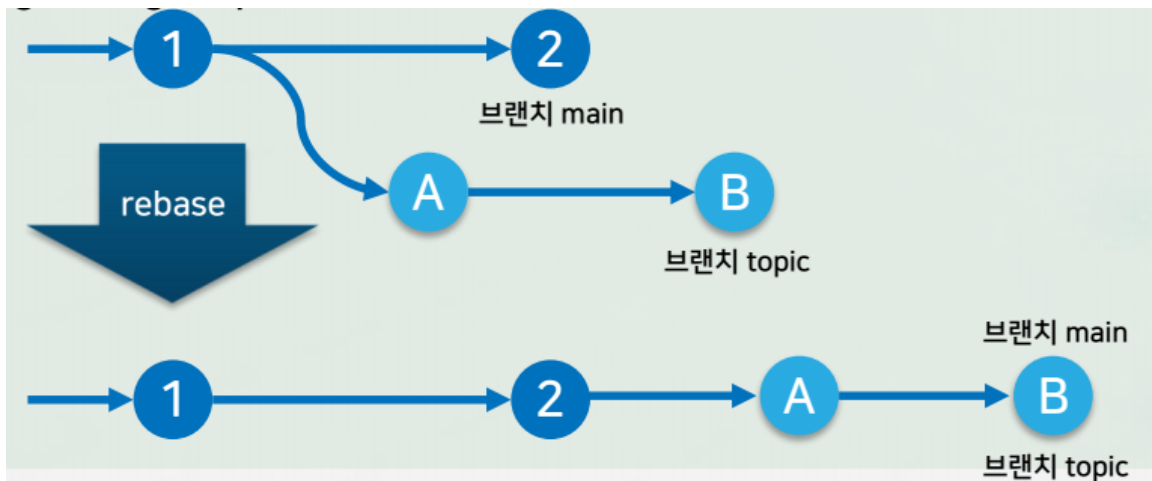


- Rebase에서의 충돌

- 충돌 발생 후 해결 절차
 - 파일 수정
 - 파일 추가
 - rebase 계속 수행 , 마지막 메시지 수정



- 기준 브랜치에서 main 브랜치 rebase 병합
 - `$ git checkout topic`
 - `$ git merge main`
- 다시 main으로 돌아와 fast-forward 병합 진행
 - `$ git checkout main`
 - `$ git merge topic`



커밋 이력 수정

최신 커밋 수정

- 최신 커밋 메시지 수정
 - `$ git commit - -amend -m '새로운 메시지'`
- 편집기로 최신 커밋 메시지 수정
 - `$ git commit - -amend`
- 파일 수정 후 추가 메시지 수정 없이 최신 커밋으로 수정
 - `$ git commit - -amend - -no-edit`
- 이전 커밋 HEAD~2..HEAD까지 각각의 커밋을 수정
 - `$ git rebase - -interactive 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
- d(rop): 커밋 자체를 삭제

비주얼스튜디오 코드의 깃 활용

- 비주얼스튜디오코드에서 깃 저장소를 생성하고 파일 생성
 - 탐색기 활동 바 , 버전 컨트롤 활동바
- 비주얼스튜디오코드에서 파일 추가, 커밋
 - Changes, staged Changes

- Changes, Staged Changes
 - 확장 Git Graph 설치와 활용
- 비주얼스튜디오코드에서 파일 비교 기능 수행
 - Changes , Staged Changes
 - Git Graph