

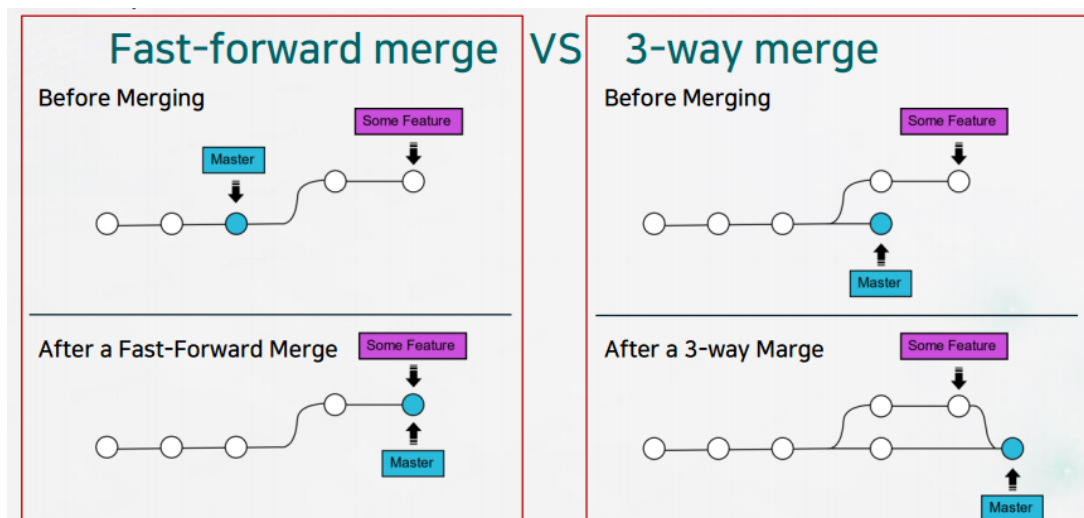


11주차

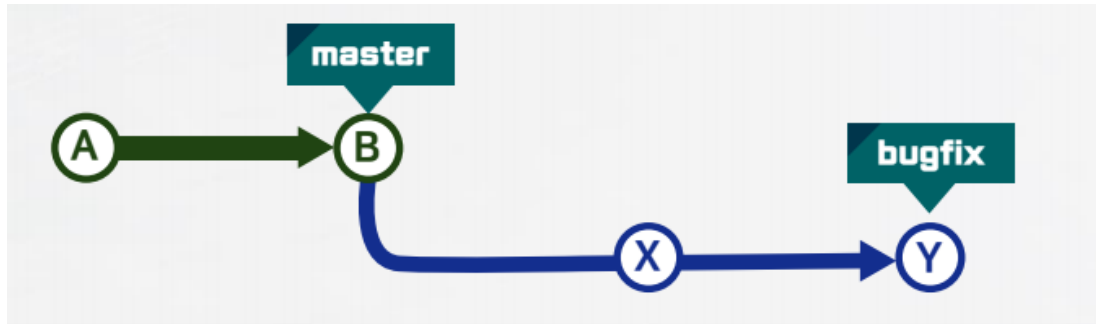
다양한 브랜치 병합

브랜치 병합

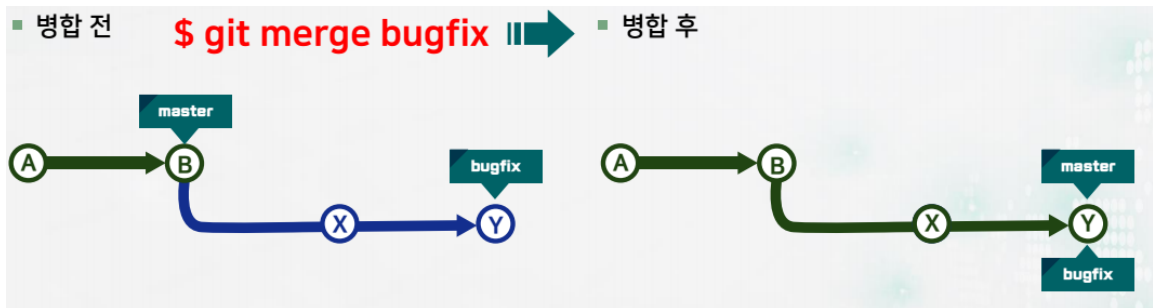
- 병합 (merge)
 - 두 개의 브랜치를 하나로 모으는 과정
 - fast-forward 병합
 - 3-way 병합



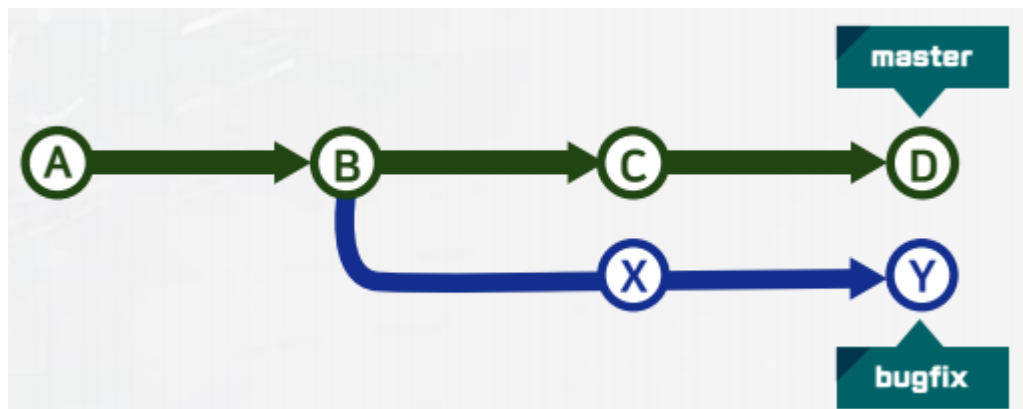
- fast -forward 병합 조건
 - 현재 브랜치 master가 병합할 대상 커밋의 직접적인 뿌리가 되는 경우
 - 간단히 두 브랜치가 일렬 상태
 - bugfix 브랜치 이력이 master 브랜치 이력을 모두 포함하는 경우



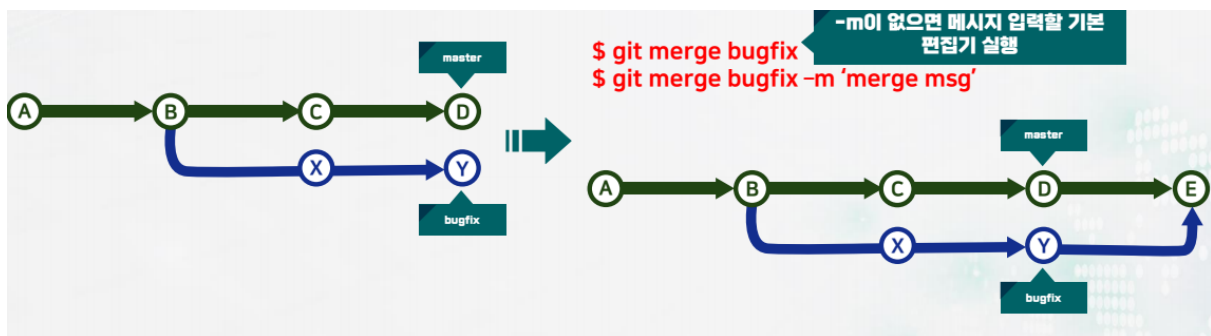
- 브랜치 master에서 병합 명령
 - `$ git merge bugfix`



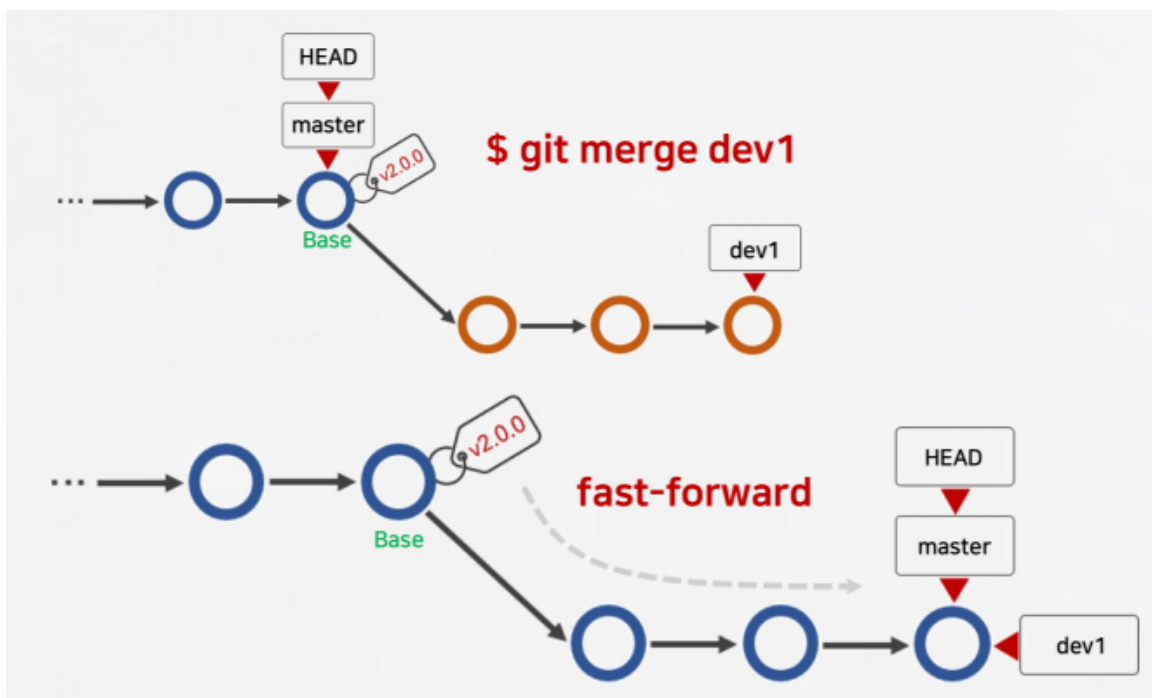
- 두 브랜치의 조상이 같은 경우
 - master 브랜치 내의 변경 내용과 bugfix 브랜치 내의 변경 내용을 하나로 통합할 필요



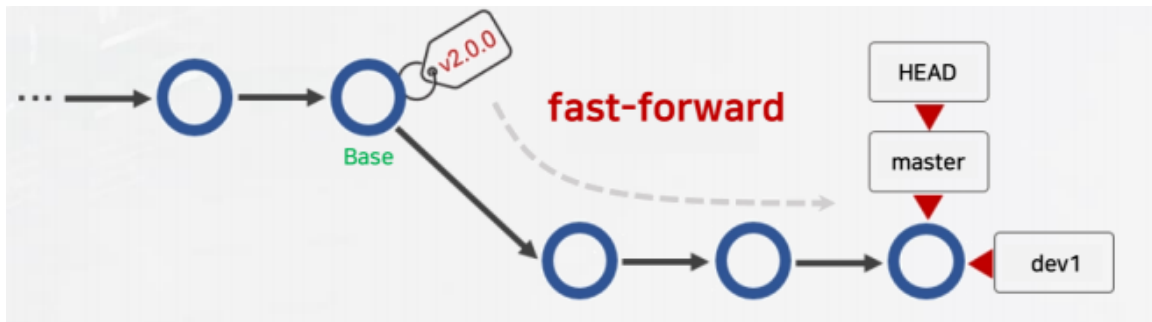
- 3-way 병합
 - 새로운 커밋을 사용하여 두 기록을 합침
- 새로운 커밋 E 생성
 - 두 브랜치의 변경을 가져온 'merge commit' E를 생성
 - 병합 완료 후 통합 브랜치인 master 브랜치로 통합된 이력이 생성
 - -m이 없으면 메시지 입력할 기본 편집기 실행



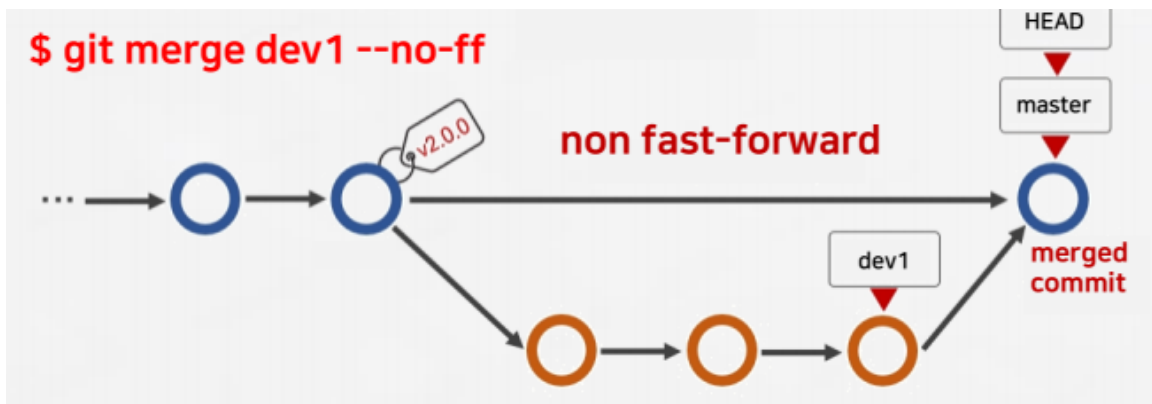
- 병합할 브랜치의 조상이 기준 브랜치인 경우 즉 일렬상태에서



- 마스터에서 dev1을 fast forward로 병합
 - `& git merge dev1`



- 마스터에서 dev1 을 non fast forward로 병합
 - 옵션 `--no-ff`
 - `$ git merge dev1 --no-ff`



- 일렬 상태

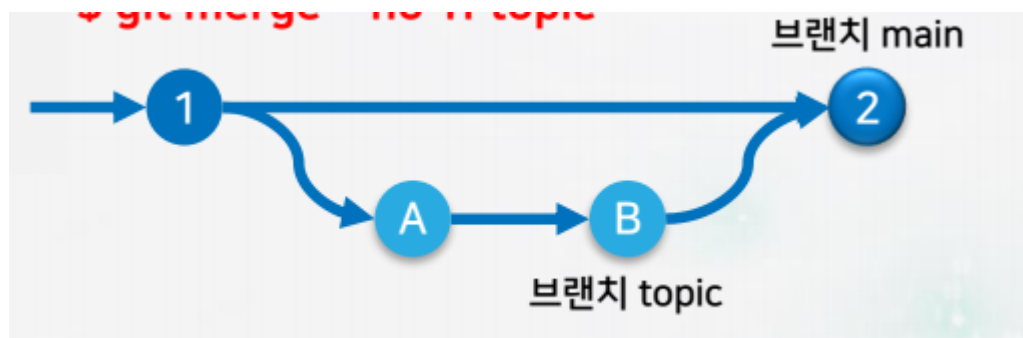


- 옵션이 있는 경우

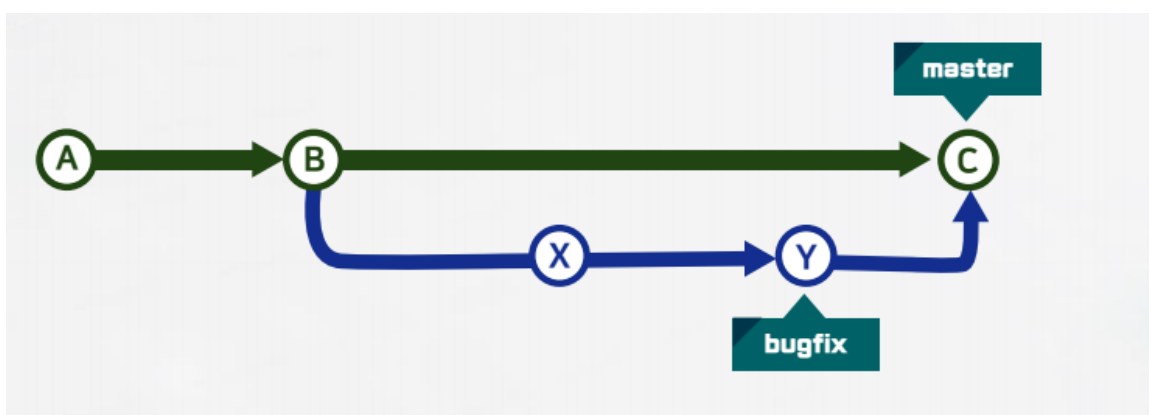
- fast-forward merge
- \$ git merge topic



- 옵션 - -no-ff
 - 3-way merge
 - \$ git merge - -no-ff-topic



- non fast-forward 병합
 - \$ git merge - -no-ff { 병합할 브랜치 명 }



- 무조건 3-way 병합 수행
 - `$ git merge - -no-ff` (병합할 브랜치 명)
- fast-forward인 경우에만 병합 진행
 - `$ git merge - -ff-only` (병합할 브랜치 명)
- 현재 브랜치에서 커밋 하나만 생성해서 병합
 - `$ git merge - -squash hotfix`
- 3-way 충돌 발생
 - `$ git merge` (브랜치 명)
- 충돌한 파일을 인지하고 파일 수정
 - `$ code file`
- 수정 후 다시 add , commit
 - `$ git commit -am ' 메시지 '`
- 충돌 이후 병합 취소
 - `$ git merge --abort`