

브랜치 병합 (merge) : 두 개의 브랜치를 하나로 모으는 과정 /fast-forward 병, 3-way 병합

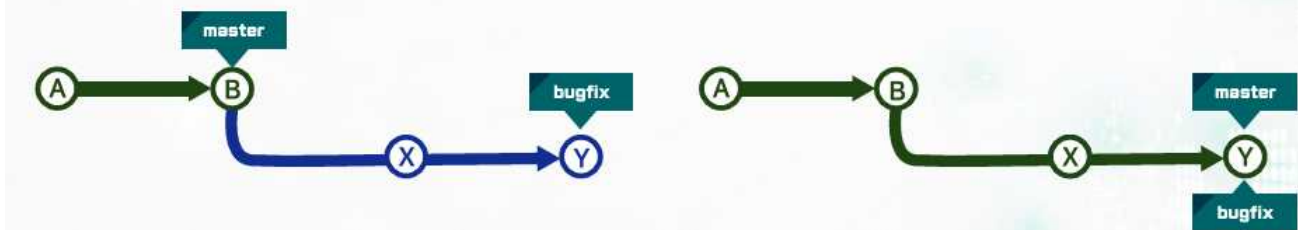
fast-forward 병합 조건

- 현재 브랜치 master가 병합할 대상 커밋의 직접적인 뿌리가 되는 경우
 - 두 브랜치가 일렬상태, bugfix 브랜치 이력이 master 브랜치 이력을 모두 포함하는 경우
- ▶ \$ git merge bugfix

- master 브랜치는 단순히 이동, 합칠 내용 X

• \$ git merge bugfix

■ 병합 전 **\$ git merge bugfix** ||| ■ 병합 후

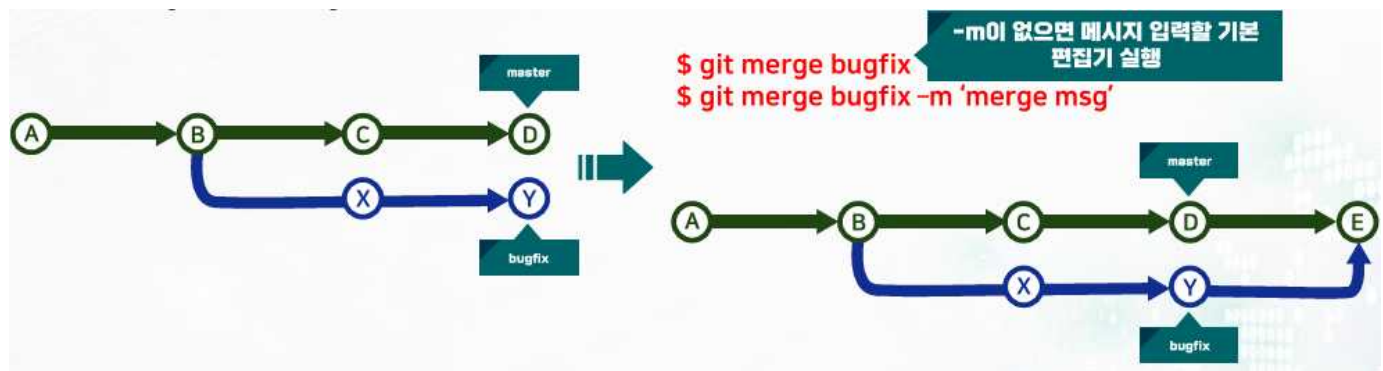


3-way 상태 : 두 분기가 갈라진 경우

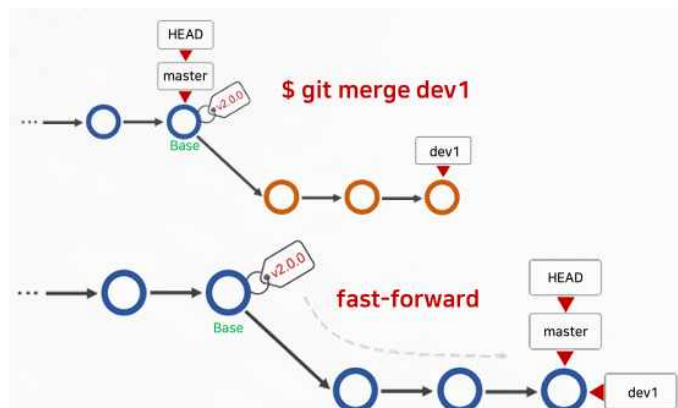
- 두 브랜치의 조상이 같은 경우
- master 브랜치 내의 변경 내용과 bugfix 브랜치 내의 변경 내용을 하나로 통합할 필요
- 새로운 커밋을 사용하여 두 기록을 합침

3-way 병합 수행

- 새로운 커밋 E 생성 후 병합

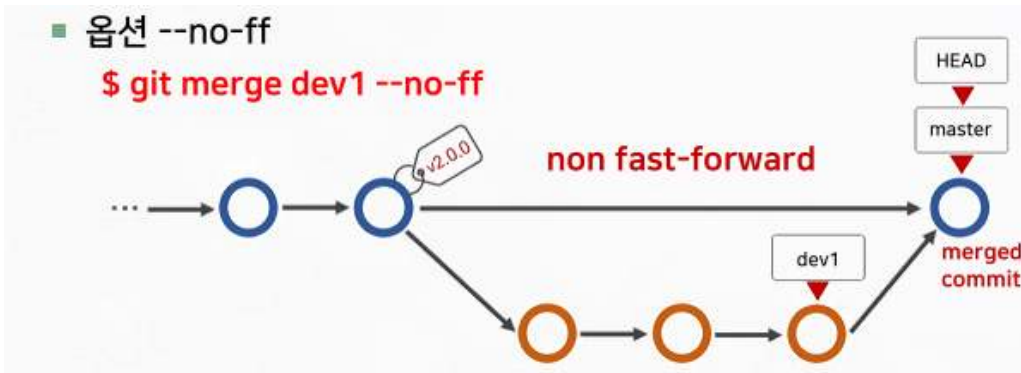


일렬상태에서 기본 fast for ward 병합 : master 브랜치에서 브랜치 dev1을 병합

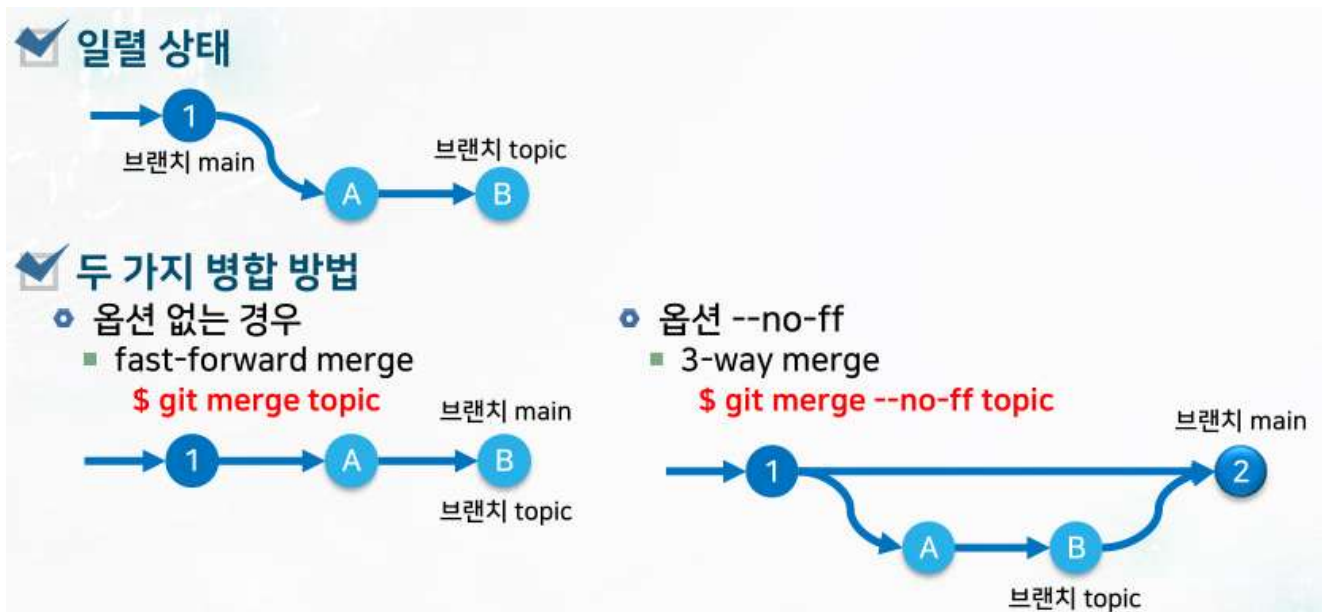


기본 fastward 상태에서 non fast forward 병합

- master에서 dev1을 fast forward로 병합 : `$ git merge dev1`
- master에서 dev1을 non fastward로 병합 (3-way 병합) : `$ git merge dev1 --no-ff`



두 브랜치의 일렬 상태에서 2가지 병합



non fast-forward 병합 옵션을 지정

- `$ git merge --no-ff {병합할 브랜치명}`
- 병합 실행 시에 fast-forward 병합이 가능한 경우라도 3-way 병합 수행
- ▶ 병합된 브랜치가 그대로 남기 때문에 그 브랜치로 실행한 작업 확인 및 브랜치 관리 면에서 유용

`$ git merge {병합할 브랜치명}`

- 보통의 병합, 융통성 있는 병합
 - fast-forward 병합 : 새로운 병합 커밋 없이 병합할 브랜치 커밋으로 이동 (일렬상태)
 - 3-way 병합 : 새로운 병합 커밋을 생성해 두 브랜치를 합침
 - `$ git merge --no-ff {병합할 브랜치명}` : 무조건 3-way 병합되는 옵션
 - `$ git merge --ff-only {병합할 브랜치명}` : 상태가 fast-forward인 상태에서만 병합 진행
 - `$ git merge --squash {병합할 브랜치명}` : 현재 브랜치에 병합 대상과의 합치는 커밋 하나 생성 후 병합
- ▶ `$ git merge --squash hotfix {병합할 브랜치명}` : 커밋 이력과 병합되는 브랜치 이력을 남기지 않음
 - 새로운 커밋에 상대 브랜치의 내용을 모두 합해 새로운 커밋으로 병합

병합 충돌 이해 (conflict)

- 3-way 상태에서 두 브랜치의 동일 조상인 커밋 1을 기준, 병합할 두 브랜치 마지막 커밋을 비교
충돌기준
- 파일 충돌 없음 : 수정되지 않거나 한쪽 브랜치에서만 수정시 : 각각 다른 파일을 수정
- 파일 충돌 발생 : 두 브랜치 모두에서 변경 사항이 발생할 시 : 파일 C는 양쪽에서 수정

충돌 해결 : 직접 파일 내용을 수정 후 저장, 계속해서 add, commit 진행

- ▶ 병합 취소 : `$ git merge --abort`
- ▶ 다시 병합 : `$ git merge feat/list`