



2강 - 커밋 변경하기

예제 1. 바로 직전 커밋 내용 수정 : `—amend` 옵션

예제 2. 이전에 작성한 커밋 지우기 : `revert`

예제 3. 쌓아온 커밋을 완전히 버리고 과거로 돌아가기 : `reset`

예제 4. 다른 브랜치의 특정 커밋을 또 다른 브랜치에 붙이기 : `cherry-pick`

예제 1. 바로 직전 커밋 내용 수정 : `—amend` 옵션

- 커밋을 작성했는데 커밋 메시지에 오타가 있네요?? 앗...
- 이럴 때는 커밋을 다시하되 `—amend` 옵션을 추가하면 바로 직전 커밋 메시지 내용을 변경할 수 있습니다.

```
$ notepad hobby.txt      // 파일 수정
$ git add --all
$ git commit -m "취미내용을 수정함ㅇㅇㅇㅋ;;;" // 오타 발생
$ git commit --amend -m "취미 내용을 수정함" // 커밋 수정
```

예제 2. 이전에 작성한 커밋 지우기 : `revert`

- 특정 커밋을 삭제하는 방법은 `reset` 명령과 `revert` 명령어가 있습니다.
- 다만 이미 깃허브와 같은 원격에 `push`된 내용은 커밋을 함부로 삭제하기가 곤란합니다.
- 이럴 때 `revert` 명령을 통해 특정 커밋을 삭제했다는 새로운 히스토리를 만들어냅니다.



출처 : https://backlog.com/git-tutorial/kr/stepup/stepup6_2.html

```
$ git switch -c feat/dream master
$ notepad dream.txt
// 첫번째 문장 작성
$ git add dream.txt
$ git commit -m "message1"
// 두번째 문장 작성
$ git commit -am "message2"
// 세번째 문장 작성
$ git commit -am "message3"
// 원격에 업로드
$ git push origin feat/dream
$ git log --oneline --all --graph
```

- 여기서 우리는 두번째 커밋을 삭제하고 싶은데 이미 깃헙에 푸시한 상태기 때문에 이력이 모든 팀원에게 공유되어 버렸기 때문에 다른 팀원이 pull을 땀겼을수도 있으므로 함부로 커밋을 지울 수 없습니다.

```
* d345578 세 번째 목표 : 킹 오브 댄스
* 172ca1e 두 번째 목표 : 자바의 왕
* 5608a90 첫 번째 목표 : 깃 정복하기
.
```

- log를 확인해 지우고 싶은 커밋의 아이디를 확인하고 revert명령을 내립니다.

```
$ git revert 172ca1e <-- 지우고 싶은 커밋 아이디
```

```
$ git revert 172ca1e
Auto-merging dream.txt
CONFLICT (content): Merge conflict in dream.txt
error: could not revert 172ca1e... 두 번째 목표 : 자바의 왕
hint: After resolving the conflicts, mark them with
hint: "git add/rm <paths>", then run
hint: "git revert --continue".
hint: You can instead skip this commit with "git revert --skip".
hint: To abort and get back to the state before "git revert",
hint: run "git revert --abort".
```

- 충돌이 났습니다. 원인은 두번째 커밋을 취소하면 첫번째 커밋상태로 돌아가겠다는 의 미인데 세번째 커밋의 처리가 걸림돌입니다.
- dream.txt를 열어줍니다.

<<<<<<< HEAD

내 첫번째 목표는 깃을 정복하는거야~

그리고 두번째 목표는 자바의 왕이 될거야~

또한 세번째 목표는 댄스킹이 되는 거지|

=====

내 첫번째 목표는 깃을 정복하는거야~

>>>>>> parent of 172ca1e (두번째 목표 : 자바의 왕)

- 원하는 형태로 수정하고 저장합니다.

내 첫번째 목표는 깃을 정복하는거야~
또한 두번째 목표는 댄스킹이 되는 거지|

- revert한 내용을 커밋합니다.

```
$ git commit -am "revert commit rollfoward to second commit"
```

```
$ git log --oneline --all --graph
* 0b558ba (HEAD -> feat/dream) reverting 두 번째 목표
* d345578 세 번째 목표 : 킹 오브 댄스
* 172ca1e 두 번째 목표 : 자바의 왕
* 5608a90 첫 번째 목표 : 깃 정복하기
```

예제 3. 쌓아온 커밋을 완전히 버리고 과거로 돌아가기 : reset

- 버전 관리 내용을 완전히 이력과 내용을 모두 잊어버리고 싶을 때 사용하는 명령어 reset입니다.
- reset은 기본옵션으로 mixed옵션이 적용되어 있으며 soft, hard, mixed 중에 선택할 수 있습니다.

모드명	HEAD의 위치	인덱스	작업 트리
soft	변경함	변경 안 함	변경 안 함
mixed	변경함	변경함	변경 안 함
hard	변경함	변경함	변경함

출처 : https://backlog.com/git-tutorial/kr/stepup/stepup6_3.html

- 예제 2에 이어서 새로운 커밋을 작성합니다.

```
$ notepad dream.txt // 텍스트 수정
$ git commit -am "message4"
```

```
$ git log --oneline --all --graph
* 9b1fe2a (HEAD -> feat/dream) 빛 나 난 빛 나 ~~
* 0b558ba reverting 두 번째 목표
* d345578 세 번째 목표 : 킹 오브 댄스
* 172ca1e 두 번째 목표 : 자 바 의 왕
* 5608a90 첫 번째 목표 : 깃 정 복 하 기
```

세번째 목표로 돌아가고 싶어~~

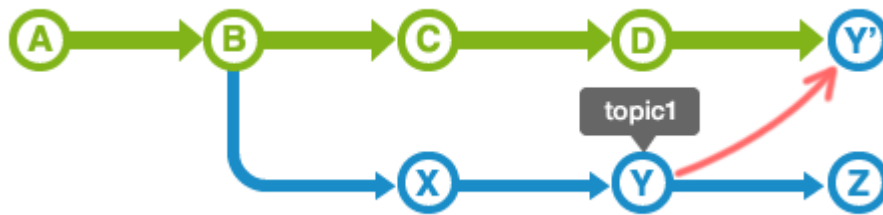
- 세번째 목표로 완전히 돌아가봅시다. 즉, 위에 2개 커밋을 완전히 날리고 싶다는겁니다.

```
// 커밋아이디를 적어도 되지만 HEAD뒤에 물결하나당 이전커밋을 가리킴
// 물결이 2개니까 2개 전 커밋으로 돌아가겠다는 뜻임
$ git reset --hard HEAD~~
$ git log --oneline --all --graph
```

```
$ git log --oneline --all --graph
* d345578 (HEAD -> feat/dream) 세 번째 목표 : 킹 오브 댄스
* 172ca1e 두 번째 목표 : 자 바 의 왕
* 5608a90 첫 번째 목표 : 깃 정 복 하 기
```

- !! 혹시 reset을 취소하고 싶으면 `\$ git reset --hard ORIG_HEAD`를 입력하세요

예제 4. 다른 브랜치의 특정 커밋을 또 다른 브랜치에 붙이기 : cherry-pick



출처 : https://backlog.com/git-tutorial/kr/stepup/stepup6_4.html

- 개발 진행 중 브랜치 이름이 develop이고 실제 사용자에게 서비스할 완성된 버전을 배포하는 브랜치 이름을 release라고 가정합니다.
- develop브랜치는 계속해서 개발을 진행하므로 커밋을 쌓아나갈 것인데, release 브랜치는 완벽하게 완성된 코드만 업로드 되어있어야 합니다.
- 이 때 develop브랜치의 수많은 커밋 중 안정화된 커밋만 똑 떼서 release브랜치에 붙여야 하는데 이 때 사용하는 명령어가 cherry-pick입니다.
- 우리 예제에서는 master브랜치를 develop브랜치로 보고, 새로운 브랜치 release를 만들어서 cherry-pick으로 특정 커밋들을 붙여보겠습니다.
- git log 명령어로 가장 최초의 커밋의 아이디를 확인하고 거기로부터 새로운 브랜치 release를 생성하고 체크아웃합니다.

```
$ git log --oneline --all --graph
$ git switch -c release 최초커밋아이디
```

```
* 0224f9a Adding diary directory and make new file into this directory
* 2ec838e Updating food.txt what the new food want to eat
* 1bf2855 Adding food.txt for the pratice1   체리픽 타겟
* a831220 Adding bye.txt for the practice1
* 4eb4bd1 (HEAD -> release) Adding hello.txt for greeting
```

```
$ git cherry-pick 1bf2855
$ git log --oneline --all --graph
```