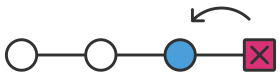
git revert

- 동작 방법
- 일반적인 옵션
- Resetting vs. reverting



git revert 명령은 '실행 취소'유형 명령으로 간주 될 수 있지만 기존 실행 취소 작업은 아닙니다. 프로젝트 히스토리에서 커밋을 제거하는 대신 커밋에 의해 도입 된 변경 사항을 반전하는 방법을 알아 내고 역순으로 새로운 커밋을 추가합니다. 이것은 Git이 히스토리를 잃어 버리는 것을 방지하며 히스토리의 무결성과 신뢰성있는 공동 작업에 중요합니다.

되돌리기는 프로젝트 내역에서 커밋의 반대를 적용하고자 할 때 사용해야합니다. 예를 들어 버그를 추적하여 단일 커밋으로 도입 된 경우에 유용합니다. 수동으로 들어가서 수정하고 새 스냅 샷을 커밋하는 대신 git 되돌리기를 사용하여 자동으로이 모든 작업을 수행 할 수 있습니다.

동작 방법

git revert 명령은 저장소의 커밋 기록에 대한 변경 사항을 실행 취소하는 데 사용됩니다. git checkout 및 git reset과 같은 다른 '실행 취소'명령은 HEAD를 이동하고 지정된 포인터로 분기 포인터를 분기합니다. Git revert는 또한 지정된 커밋을 취하지 만, git revert는이 커밋에 대한 ref 포인터를 이동시키지 않습니다. 되돌리기 작업은 지정된 커밋을 취하여 해당 커밋에서 변경 사항을 역으로 적용하고 새로운 "되돌리기 커밋"을 만듭니다. 그런 다음 ref 포인터는 새로운 복귀 커밋을 가리 키도록 업데이트되어 지점의 끝으로 지정됩니다.

```
$ mkdir git_revert_test
$ cd git_revert_test/
$ git init .
Initialized empty Git repository in /git_revert_test/.git/
$ touch demo_file
$ git add demo_file
$ git commit -am"initial commit"
[master (root-commit) 299b15f] initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 demo_file
$ echo "initial content" >> demo_file
$ git commit -am"add new content to demo file"
[master 3602d88] add new content to demo file
n 1 file changed, 1 insertion(+)
$ echo "prepended line content" >> demo_file
$ git commit -am"prepend content to demo file"
[master 86bb32e] prepend content to demo file
 1 file changed, 1 insertion(+)
$ git log --oneline
86bb32e prepend content to demo file
3602d88 add new content to demo file
299b15f initial commit
```

여기서는 git_revert_test라는 새로 생성 된 디렉토리에서 repo를 초기화했습니다. demo_file 파일을 추가하고 내용을 두 번 수정 한 repo에 대해 3 번 커밋했습니다. repo 설정 절차가 끝나면 git log를 호출하여 커밋 기록을 표시하고 총 3 개의 커밋을 표시합니다. 이 상태의 repo를 사용하여 git 되돌리기를 시작할 준비가되었습니다.

```
$ git revert HEAD
[master b9cd081] Revert "prepend content to demo file"
1 file changed, 1 deletion(-)
```

Git Revert는 커밋 참조가 전달되고 전달되지 않고 실행되지 않을 것으로 기대합니다. 여기서 우리는 HEAD ref를 통과했다. 이렇게하면 최신 커밋을 되돌릴 수 있습니다. 이는 3602d8815dbfa78cd37cd4d189552764b5e96c58을 실행하기 위해 되돌릴 때와 동일한 동작입니다. 병합과 마찬가지로 되돌리기는 새 커밋 메시지를 요구하는 구성된 시스템 편집기를 열 수있는 새로운 커밋을 만듭니다. 커밋 메시지가 입력되고 저장되면 힘내 기능이 작동을 재개합니다. 이제 git log를 사용하여 repo의 상태를 검사하고 이전 로그에 새로운 커밋이 추가되었음을 확인할 수 있습니다.

Git 가이드

1. Git 시작하기
   - Git 저장소
     - git init
     - git clone
     - git config
   - 변경 저장하기
     - git add
     - git commit
     - git diff
     - git stash
     - .gitignore
   - 저장소 점검하기
     - git status
     - git log
     - git tag
     - git blame
   - 변경 취소하기
     - git 실행 취소
     - git clean
     - git revert
     - git reset
     - git rm
   - Rewriting history
     - git commit --amend
     - git rebase
     - git reflog
2. Git 협업하기
   - 동기화하기
     - git remote
     - git fetch
     - git pull
     - git push
   - 브랜치 사용하기
     - git branch
     - git checkout
     - git merge
     - 병합 충돌 해결하기 (Merge conflicts)
     - 병합 전략 (Merge strategies)
   - Pull request 만들기

```
$ git log --oneline
1061e79 Revert "prepend content to demo file"
86bb32e prepend content to demo file
3602d88 add new content to demo file
299b15f initial commit
```

세 번째 커밋은 되돌리기 후에도 프로젝트 기록에 남아 있습니다. 삭제하는 대신 변경 사항을 취소하기위한 새로운 커밋이 추가되었습니다. 결과적으로 두 번째와 네 번째 커밋은 정확히 같은 코드 기반을 나타내며 세 번째 커밋은 우리가 길을 따라 다시 돌아가고 싶은 경우에 대비하여 우리 역사에 남아 있습니다.

일반적인 옵션

```
-e
--edit
```

이 옵션은 기본 옵션이므로 지정하지 않아도됩니다. 이 옵션은 구성된 시스템 편집기를 열고 되돌리기를 커밋하기 전에 커밋 메시지를 편집하라는 메시지를 표시합니다.

```
--no-edit
```

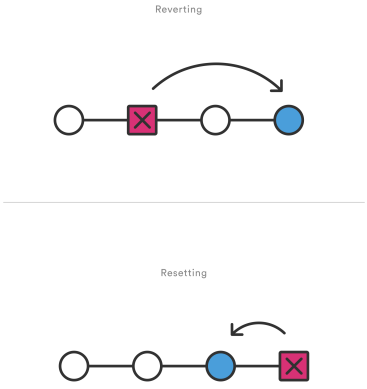
이것은 -e 옵션의 반대입니다. 되돌리기는 편집기를 열지 않습니다.

```
-n
--no-commit
```

이 옵션을 건네면 git revert가 타겟 커밋을 반대로하는 새로운 커밋을 만드는 것을 막을 수 있습니다. 새 커밋을 만드는 대신이 옵션을 사용하면 준비 인덱스 및 작업 디렉터리에 역 변경 사항이 추가됩니다. 이것은 Git이 저장소의 상태를 관리하는 데 사용하는 다른 트리입니다. 자세한 정보는 git reset 페이지를 참조하십시오.

Resetting vs. reverting

git revert가 단일 커밋을 취소한다는 것을 이해하는 것이 중요합니다. 모든 커밋을 제거함으로써 프로젝트의 이전 상태로 "되돌아 가지"않습니다. 힘내 (Git)에서는 이것을 실제로 리셋 (reset)이라고하며 되돌리기 (revert)라고 부르지 않습니다.



되돌리기에는 재설정보다 두 가지 중요한 이점이 있습니다. 첫째, 공유 저장소에 이미 게시 된 커밋에 대해 "안전한"작업을 수행하는 프로젝트 기록을 변경하지 않습니다. 공유 기록을 변경하는 것이 위험한 이유에 대한 자세한 내용은 [git reset] 페이지를 참조하십시오.

둘째, git revert는 히스토리의 임의의 지점에서 개별 커밋을 대상으로 할 수 있지만 git reset은 현재 커밋에서 뒤로 만 작동 할 수 있습니다. 예를 들어 git reset으로 이전 커밋을 실행 취소하려면 대상 커밋 이후에 발생한 모든 커밋을 제거하고 제거한 다음 모든 커밋을 다시 커밋해야합니다. 말할 필요도없이, 이것은 우아한 실행 취소 솔루션이 아닙니다. git 되돌리기와 다른 '실행 취소'명령의 차이점에 대한 자세한 설명은 재설정, 체크 아웃 및 되돌리기를 참조하십시오.