







- revert HEAD
- 2. revert --no-edit HEAD
- 3. revert --abort



- 1. 특정 커밋 이력 취소하고 바로 이전 상태로 돌아가는 방법을 수행할 수 있다.
- 2. 명령 revert에서 자동으로 새로운 커밋 메시지를 생성할 수 있다.
- 3. 버전 되돌리기 reset과 취소 revert의 차이를 이해하고 활용할 수 있다.







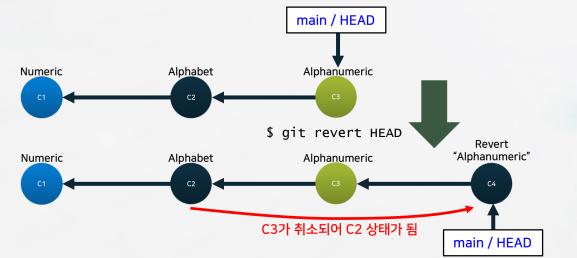




### 🦈 커밋 취소 revert

### ₩ undo와 비슷한 기능

- 지정한 특정 커밋을 취소해 바로 이전 상태로 되돌리는 방법
- reset과 다르게 커밋할 것이 없고, 작업 영역이 깨끗해야 수행 가능
  - Nothing to commit. working tree clean
  - 커밋해온 모든 변경 사항들을 rollback
  - 이전의 커밋 히스토리는 그대로 유지, 되돌리는 새로운 커밋이 그 이후에 추가





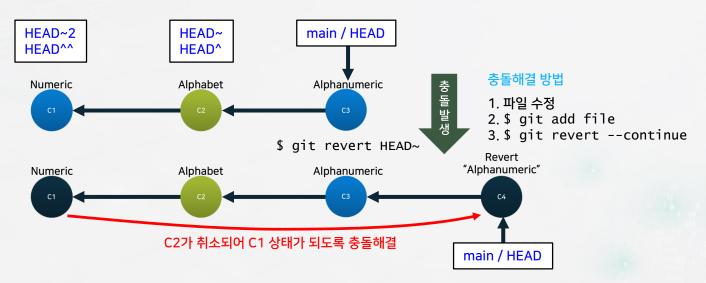


### 커밋취소 revert

### 🌣 revert 충돌과 해결

### ✓ 인자가 HEAD~와 같이 HEAD 이전을 취소하면 충돌이 발생할 수 있음

- 충돌이 발생하지 않으려면
  - 바로 이전 취소를 여러 번 계속
  - 마치 워드 프로세서에서 취소 [ctrl + z]를 여러 번 하는 것과 비슷







### 커밋취소revert

### 명령 revert



옵션 --no-edit

○ 추가되는 커밋 메시지가 자동으로 'Revert "이전 커밋 메시지"'으로 지정

주요 명령	
<pre>\$ git revert HEAD~ \$ git revert [commit_ID]</pre>	이동되는 커밋의 내용으로 작업 폴더와 스테이지 영역, 깃 저장소를 모두 복사하고 커밋 메시지를 수정하도록 편집기가 실행됨
<pre>\$ git revert HEAD~no-edit \$ git revert [commit_ID]no-edit</pre>	옵션no-edit으로 편집기 실행 없이 커밋 메시지가 자동으로 'Revert "이전 커밋 메시지"'으로 지정되어 새로운 커밋 생성





### 1 커밋취소 revert

### 🗘 오류가 발생해 revert가 취소

- ▼ 커밋할 것이 있고 작업 영역(working tree)이 깨끗(clean)하지 않으면 오류
  - Nothing to commit, working tree clean

### ₩ 해결 방법

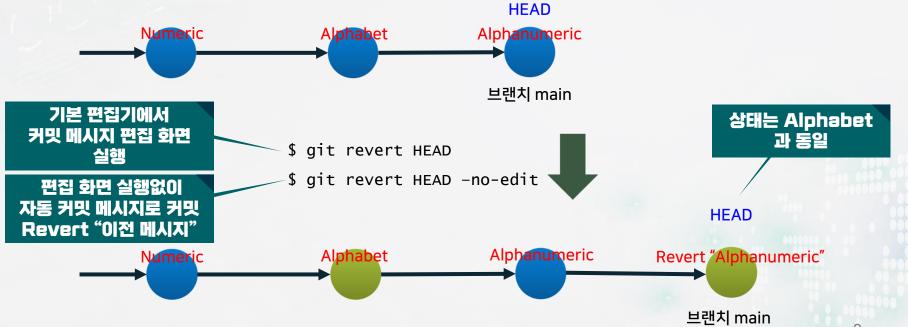
- 커밋을 하고 다시 revert 하는 방법
  - 현재 커밋 내용으로 작업 폴더와 스테이지 영역, 깃 저장소를 모두 수정·복사
    - ➡ 수정한 내용이 사라짐
- git reset --hard HEAD~
  - 이전 커밋 내용으로 작업 폴더와 스테이지 영역, 깃 저장소를 모두 수정·복사
    - ➡ 수정한 내용이 사라지고
    - ➡ 현재 커밋 로그도 삭제





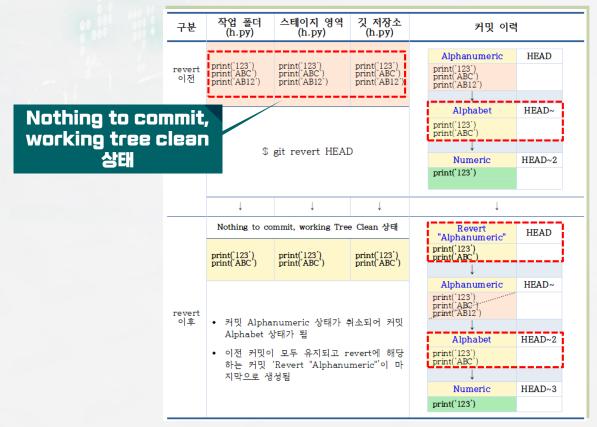
### revert로 HEAD 바로 이전 상태로 돌아가기

- ★ 취소 인자를 HEAD로 사용
  - 이전 커밋 수 3개에 비해 revert 성공 후 커밋이 하나 추가되어 총 4개



### 커밋취소revert

### 🐡 \$ git revert HEAD









### (실습) 3번의 커밋

## ☑ 이전의 자료를 간략히

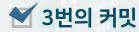
\$ git revert HEAD

\$ git revert HEAD --no-edit

구분	작업 폴더 (h.py)	스테이지 영역 (h.py)	깃 저장소 (h.py)	커밋	이력
revert 이전	123 ABC AB12	123 ABC AB12	123 ABC AB12	Alphanumeric  123 ABC AB12	HEAD
	\$ git revert HEAD			Alphanumeric 123 ABC	HEAD~
				Numeric 123	HEAD~2







- \$ git init grvt
  \$ cd grvt
- \$ echo 123 > h
- \$ git add h
- \$ git commit -m Numeric
- \$ echo ABC >> h
- \$ git commit -am Alphabet
- \$ echo AB12 >> h
- \$ git commit -am Alphanumeric
- \$ cat h
- 123
- ABC
- AB12







### 커밋취소 revert



### ☑ 이전 커밋과 비교

```
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git log --oneline
0a4b9ea (HEAD -> main) Alphanumeric
4f760d4 Alphabet
0382f14 Numeric
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git diff HEAD
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git diff HEAD~
diff --git a/h b/h
index 09a66f1..88869c9 100644
--- a/h
+++ b/h
@@ -1,2 +1,3 @@
123
 ABC
+AB12
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git diff HEAD~2
diff --git a/h b/h
index 190a180..88869c9 100644
--- a/h
+++ b/h
@@ -1 +1,3 @@
123
+ABC
+AB12
```





### 🐞 \$ git revert HEAD --no-edit

### ☑ 자동으로 취소 커밋 메시지 생성

Revert "Alphanumeric"

```
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git log --oneline
Oa4b9ea (HEAD -> main) Alphanumeric
4f760d4 Alphabet
0382f14 Numeric
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git revert HEAD --no-edit
[main da0abb0] Revert "Alphanumeric"
 Date: Tue Jan 24 13:39:07 2023 +0900
 1 file changed, 1 deletion(-)
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git log --oneline
daOabbO (HEAD -> main) | Revert "Alphanumeric"
0a4b9ea Alphanumeric
4f760d4 Alphabet
0382f14 Numeric
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ cat h
123
ABC
```

### 커밋취소revert



### 🐞 바로 전 revert 취소

## **★** \$ git reset --hard HEAD~

```
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git log --oneline
daOabbO (HEAD -> main) Revert "Alphanumeric"
Oa4b9ea Alphanumeric
4f76Od4 Alphabet
0382f14 Numeric
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ cat h
123
ABC
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git reset --hard HEAD~!
HEAD is now at 0a4b9ea Alphanumeric
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git log --oneline
Oa4b9ea (HEAD -> main) Alphanumeric
4f760d4 Alphabet
0382f14 Numeric
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ cat h
123
ABC
AB12
```



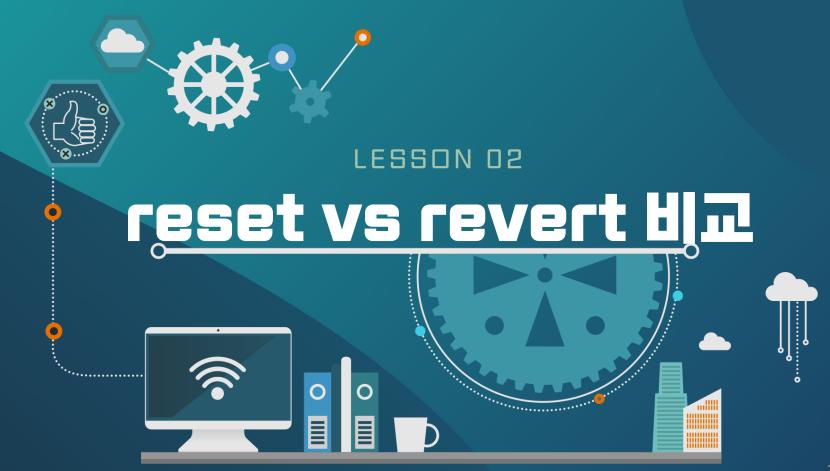


### 🐡 \$ git revert HEAD

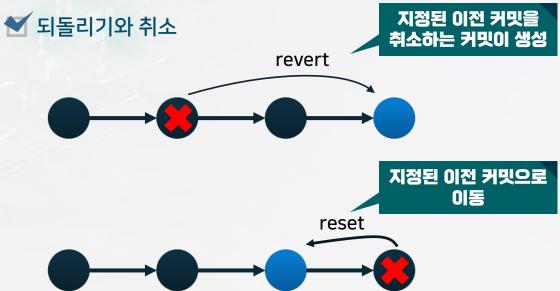
### ☑ 기본 편집기로 새로운 취소 메시지 편집 후 저장, 닫기

```
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git log --oneline
<u>0a4b9ea (HEAD -> main)</u> Alphanumeric
4f760d4 Alphabet
0382f14 Numeric
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git revert HEAD
main 232a1c9] Revert "Alphanumeric" to "Alphabet"
 1 file changed, 1 deletion(-)
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ git log --oneline
232a1c9 (HEAD -> main) Revert "Alphanumeric" to "Alphabet"
Oa4b9ea Alphanumeric
4f760d4 Alphabet
0382f14 Numeric
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/grvt (main)
$ cat h
123
ABC
```





- 2 reset vs revert 비교
- reset vs revert







### 2 reset vs revert 出교

### reset vs revert

- ☑ 되돌리기, 재설정 reset
  - 특정 커밋으로 되돌아가고 그 이후의 로그 이력은 모두 제거
- 對 커밋 취소 revert
  - 특정 커밋을 취소하는 새로운 커밋 생성, 이전 모든 이력은 그대로 남음

항목	reset	revert
기능	이전 커밋 상태로 이동	이전 커밋을 취소해 그 전 상태로 이동
새로운 커밋	없음	있음
이전 커밋 이력	삭제됨	유지됨
커밋할 것 없고 작업 트리 클린 (Nothing to commit, working tree clean)	상관없음	깨끗해야 함 (Nothing to commit, working tree clean)
충돌 발생	없음	발생할 수 있음
충돌 해결	-	add 후 commit 또는continue
옵션	hard mixed soft	continue abort no-edit -n  no-commit





# Summary

- >> 취소 revert 전체 조건
  - Nothing to commit, working tree clean
- >> HEAD를 취소해 HEAD~ 상태로 가는 커밋을 생성
  - \$ git revert HEAD
  - ◆ \$ git revert HEAD --no-edit