

Open source Software

오픈소스 소프트웨어

28. 버전 되돌리기 reset



학습 개요

1. 과거 버전으로 되돌리기
2. reset 옵션 --hard --mixed --soft
3. 작업 디렉토리(폴더), 스테이징 영역, 깃 저장소



학습 목표

1. 과거 버전으로 완전히 되돌리는 방법을 이해할 수 있다.
2. 버전 되돌리는 reset 명령을 수행할 수 있다.
3. reset과 checkout 차이를 이해하고 활용할 수 있다.

LESSON 01

버전 되돌리기 reset과 옵션

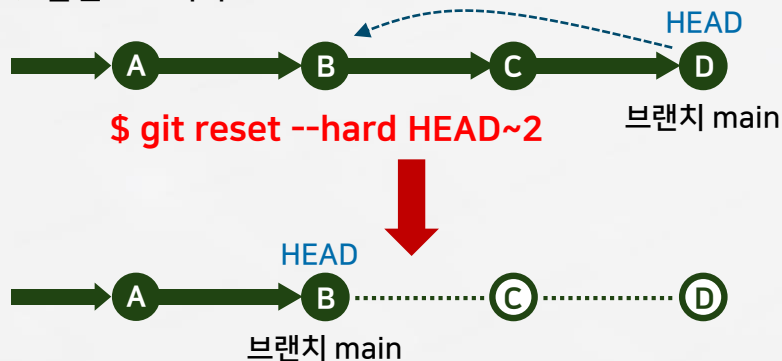


1 버전 되돌리기 reset과 옵션

기능 reset 이해

✓ 커밋 이력에서 이전 특정 커밋으로 완전히 되돌아가는(roll back) 방법

- 시계를 뒤로 맞추는 '타임 머신'과도 같음
 - 이동되는 해당 커밋 이후의 이력은 모두 사라지므로 주의가 필요
 - 새로운 커밋이 생성되지 않음
 - 깃 저장소는 이전 커밋 내용으로 수정
 - 다만 reset 이전에 있던 작업 폴더와 스테이지 영역을 내용을 어떻게 유지할지가 관건
- ➡ 옵션으로 처리



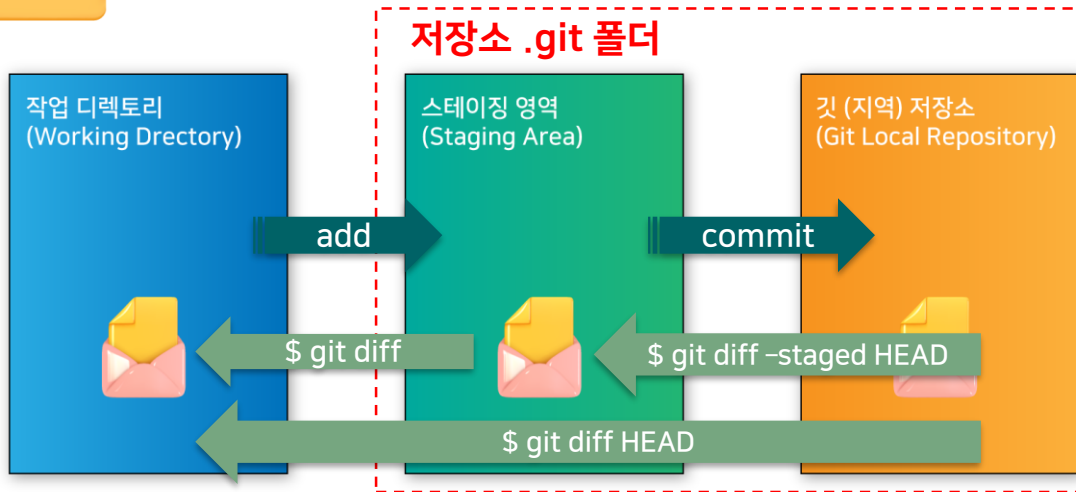
1 버전 되돌리기 reset과 옵션

기어 아이콘 3 영역

- 작업 디렉토리(폴더), 스테이징 영역, 깃 저장소



탐색기 저장소 폴더



1 버전 되돌리기 reset과 옵션

버전 되돌리기 reset

준비, 현재 상태

작업 폴더 (h.py)	스테이지 영역 (h.py)	깃 저장소 (h.py)	커밋 이력												
<pre>print('123') print('ABC') print('AB12') print([1, 2]) print({1, 2})</pre>	<pre>print('123') print('ABC') print('AB12') print([1, 2])</pre>	<pre>print('123') print('ABC') print('AB12')</pre>	<table><tr><td>Alphanumeric</td><td>HEAD</td></tr><tr><td>print('123') print('ABC') print('AB12')</td><td></td></tr></table> <p>↓</p> <table><tr><td>Alphanumeric</td><td>HEAD~</td></tr><tr><td>print('123') print('ABC')</td><td></td></tr></table> <p>↓</p> <table><tr><td>Numeric</td><td>HEAD~2</td></tr><tr><td>print('123')</td><td></td></tr></table>	Alphanumeric	HEAD	print('123') print('ABC') print('AB12')		Alphanumeric	HEAD~	print('123') print('ABC')		Numeric	HEAD~2	print('123')	
Alphanumeric	HEAD														
print('123') print('ABC') print('AB12')															
Alphanumeric	HEAD~														
print('123') print('ABC')															
Numeric	HEAD~2														
print('123')															

1 버전 되돌리기 reset과 옵션

⚙️ reset 옵션 --hard, --mixed, --soft

✓ reset의 3가지 옵션

- 무조건 깃 저장소에는 복사
- 작업 폴더와 스테이지 영역의 내용은 어떻게 처리할까?
 - --hard, --mixed, --soft

주요 명령	옵션 --mixed는 옵션이 없는 것과 동일
\$ git reset --hard HEAD~ \$ git reset --hard [commit_ID]	이동되는 커밋의 내용으로 작업 폴더와 스테이지 영역, 깃 저장소를 모두 복사
\$ git reset HEAD~ \$ git reset [commit_ID] \$ git reset --mixed HEAD~ \$ git reset --mixed [commit_ID]	이동되는 커밋의 내용으로 스테이지 영역과 깃 저장소 두 부분에 복사, 작업 폴더는 이전 내용 그대로 남음
\$ git reset --soft HEAD~ \$ git reset --soft [commit_ID]	이동되는 커밋의 내용으로 깃 저장소에만 복사, 스테이지 영역과 작업 폴더는 이전 내용 그대로 남음

1 버전 되돌리기 reset과 옵션

reset 옵션 --hard

\$ git reset --hard HEAD~

- 지정된 HEAD~(커밋 메시지 Alphanumeric)의 내용으로 작업 폴더와 스테이지 영역, 깃 저장소가 모두 복사·수정
 - reset 전에 있던 작업 폴더와 스테이지 영역에 작업 내용이 모두 사라지므로 주의가 필요

구분	작업 폴더 (h.py)	스테이지 영역 (h.py)	깃 저장소 (h.py)	커밋 이력	
reset 이전	print('123') print('ABC') print(' AB12') print([1, 2]) print({1, 2})	print('123') print('ABC') print('AB12') print([1, 2])	print('123') print('ABC') print('AB12')	Alphanumeric	HEAD
				print('123') print('ABC') print('AB12')	
↓	\$ git reset --hard HEAD~				
	↓	↓	↓	Alphabet	HEAD~
reset 이후	print('123') print('ABC')	print('123') print('ABC')	print('123') print('ABC')	print('123') print('ABC')	
				Numeric	HEAD~2
				print('123')	

1 버전 되돌리기 reset과 옵션

reset 옵션 --hard

\$ git reset --hard HEAD~2

- 지정된 HEAD~(커밋 메시지 Alphanumeric)의 내용으로 작업 폴더와 스테이지 영역, 깃 저장소가 모두 복사·수정
 - reset 전에 있던 작업 폴더와 스테이지 영역에 작업 내용이 모두 사라지므로 주의가 필요

구분	작업 폴더 (h.py)	스테이지 영역 (h.py)	깃 저장소 (h.py)	커밋 이력	
reset 이전	print('123') print('ABC') print('AB12') print([1, 2]) print({1, 2})	print('123') print('ABC') print('AB12') print([1, 2])	print('123') print('ABC') print('AB12')	Alphanumeric	HEAD
				print('123') print('ABC') print('AB12')	
				↓	
↓ \$ git reset --hard HEAD~2 ↓	↓	↓	↓	Alphabet	HEAD~
				print('123') print('ABC')	
				↓	
reset 이후	print('123')	print('123')	print('123')	Numeric	HEAD~2
				print('123')	

1 버전 되돌리기 reset과 옵션

reset 옵션 --mixed

\$ git reset --mixed HEAD~

- 지정된 HEAD~(커밋 메시지 Alphanumeric)의 내용으로 스테이지 영역과 깃 저장소가 모두 복사·수정
 - 커밋 메시지 Alphanumeric의 로그 이력과 함께 당시의 스테이지 영역, 깃 저장소 내용이 모두 사라짐
 - 다만 작업 폴더의 내용은 이전 그대로 남음
- mixed는 기본 옵션으로 git reset과 동일
 - 옵션 --mixed는 옵션이 없는 것과 같음

구분	작업 폴더 (h.py)	스테이지 영역 (h.py)	깃 저장소 (h.py)	커밋 이력	
reset 이전	print('123') print('ABC') print('AB12') print([1, 2]) print({1, 2})	print('123') print('ABC') print('AB12') print([1, 2]) print({1, 2})	print('123') print('ABC') print('AB12')	Alphanumeric	HEAD
↓	\$ git reset --mixed HEAD~			print('123') print('ABC') print('AB12')	
	↓	↓	↓	Alphabet	HEAD~
reset 이후	print('123') print('ABC') print('AB12') print([1, 2]) print({1, 2})	print('123') print('ABC')	print('123') print('ABC')	print('123') print('ABC')	
				Numeric	HEAD~2
				print('123')	

1 버전 되돌리기 reset과 옵션

reset 옵션 soft

\$ git reset --soft HEAD~

- 지정된 HEAD~(커밋 메시지 Alphanumeric)의 내용으로 깃 저장소만 복사·수정
- 커밋 메시지 Alphanumeric의 로그 이력은 사라짐
 - 작업 폴더와 스테이지 영역의 내용이 모두 이전 그대로 남음

구분	작업 폴더 (h.py)	스테이지 영역 (h.py)	깃 저장소 (h.py)	커밋 이력	
reset 이전	print('123') print('ABC') print('AB12') print([1, 2]) print({1, 2})	print('123') print('ABC') print('AB12') print([1, 2])	print('123') print('ABC') print('AB12')	Alphanumeric	HEAD
↓	\$ git reset --soft HEAD~			print('123') print('ABC') print('AB12')	
	↓	↓	↓	Alphabet	HEAD~
	↓	↓	↓	print('123') print('ABC')	
reset 이후	print('123') print('ABC') print('AB12') print([1, 2]) print({1, 2})	print('123') print('ABC') print('AB12') print([1, 2])	print('123') print('ABC')	Numeric	HEAD~2
				print('123')	



LESSON 02

reset 정리, checkout과 reset 비교

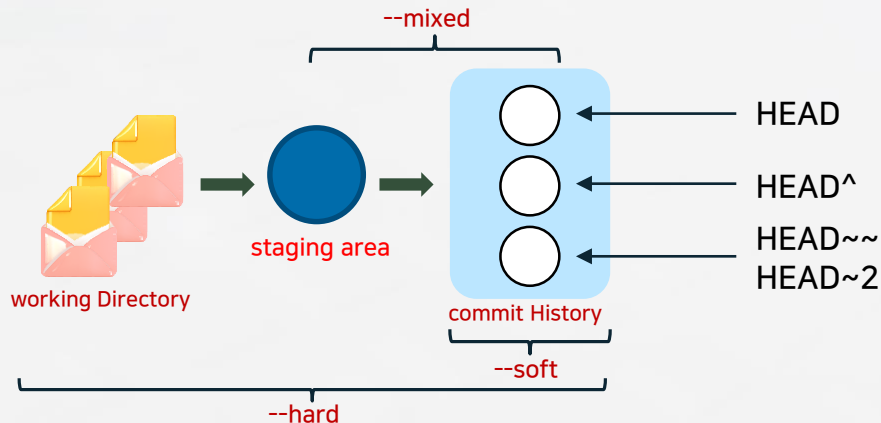


2 reset 정리, checkout과 reset 비교

reset 3가지 방식

\$ git reset --옵션 commitID

- hard
 - 인자인 커밋 깃 저장소의 내용을 작업 폴더와 스테이지 영역, 그리고 깃 저장소 모두에 복사·수정
- mixed
 - 기본 옵션으로 스테이지 영역과 깃 저장소 두 곳에 복사·수정
- soft
 - 깃 저장소에만 복사·수정하므로 작업 폴더와 스테이지 영역은 이전 내용이 그대로 남음



2 reset 정리, checkout과 reset 비교

reset 옵션 정리

옵션 --hard --mixed --soft

구분	작업 폴더 (h.py)	스테이지 영역 (h.py)	깃 저장소 (h.py)	커밋 이력	
reset 이전	print('123') print('ABC') print('AB12') print([1, 2]) print({1, 2})	print('123') print('ABC') print('AB12') print([1, 2])	print('123') print('ABC') print('AB12')		
↓	\$ git reset --옵션 HEAD~			Alphanumeric	HEAD
	↓	↓	↓	print('123') print('ABC') print('AB12')	
reset 이후	--hard	print('123') print('ABC')	print('123') print('ABC')	Alphabet	HEAD~
	--mixed 기본 옵션	print('123') print('ABC') print('AB12') print([1, 2]) print({1, 2})	print('123') print('ABC')	print('123') print('ABC')	
	--soft	print('123') print('ABC') print('AB12') print([1, 2]) print({1, 2})	print('123') print('ABC') print('AB12')	Numeric	HEAD~2
				print('123')	

2 reset 정리, checkout과 reset 비교

커밋 되돌리기

✓ \$ git reset <옵션> <돌아가고 싶은 커밋>

- hard, mixed, soft 세가지
- --soft
 - 돌아가려 했던 이력으로 되돌아 갔지만
 - 해당 내용의 인덱스(또는 스테이지)도 이전 그대로
 - ➡ 바로 다시 커밋할 수 있는 상태로 남아있는 것
 - 다시 마지막 이전 head로 돌아가려면
 - ➡ Commit 만 필요
- --mixed
 - 역시 이력은 되돌려지고, 인덱스도 되돌아가는 커밋의 내용으로 초기화
 - 다시 마지막 이전 head로 돌아가려면
 - ➡ add, commit이 필요
- --hard
 - 돌아가려는 이력 이후의 모든 내용을 삭제
 - 다시 마지막 이전 head로 돌아가려면
 - ➡ 파일 수정, add, commit 필요

2 reset 정리, checkout과 reset 비교

reset 후 다시 바로 이전 상태로 되돌아가기

'ORIG_HEAD' 이용하면 매우 간단

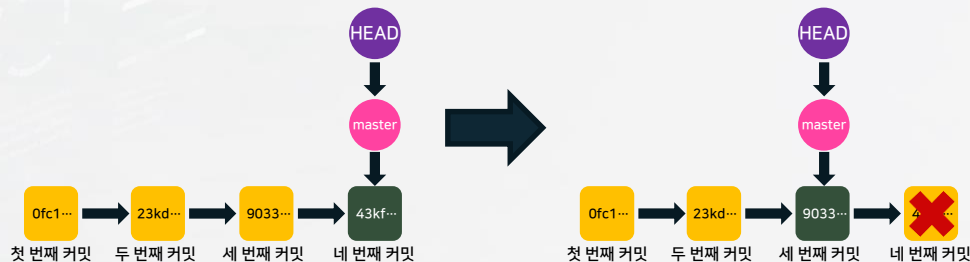
- 바로 이전 커밋 참조
 - \$ git reset --hard ORIG_HEAD

2 reset 정리, checkout과 reset 비교

reset과 checkout 비교

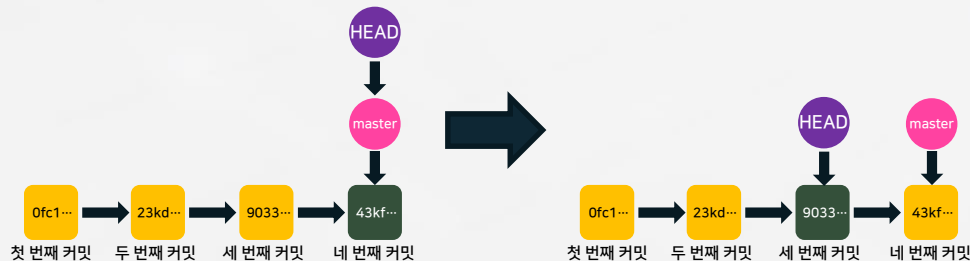
\$ git reset 9033

- 브랜치의 마지막 커밋을 수정하는 명령



\$ git checkout 9033

- HEAD 포인터를 브랜치 마지막 커밋 이전으로 이동하는 명령



Summary

» HEAD~2의 내용으로 작업 디렉토리와 스테이징 영역, 깃 저장소에 복사

- ◆ \$ git reset --hard HEAD~2

» HEAD~2의 내용으로 스테이징 영역과 깃 저장소에 복사

- ◆ \$ git reset --mixed HEAD~2

» HEAD~2의 내용으로 깃 저장소에 복사

- ◆ \$ git reset --soft HEAD~2

» 이전에 수행한 reset을 바로 취소하는 명령

- ◆ \$ git reset --hard ORIG_HEAD