

# Git reset과 revert

Version Control and  
Git & Github basics

강환수 교수



- 커밋 이력을 되돌리는 방법을 이해하고 수행할 수 있다.
  - Reset
    - 커밋 이력 재설정을 수행할 수 있다.
  - revert
    - 특정 커밋 이력 취소하고 바로 이전 상태로 돌아가는 방법을 수행할 수 있다.
  - Reset vs Revert
    - Reset은 되돌리는 이후 커밋 이력이 제거되고
    - Revert는 현재까지의 커밋 이력에 되돌리기 커밋이 추가됨

AI Experts  
Who Lead  
The Future

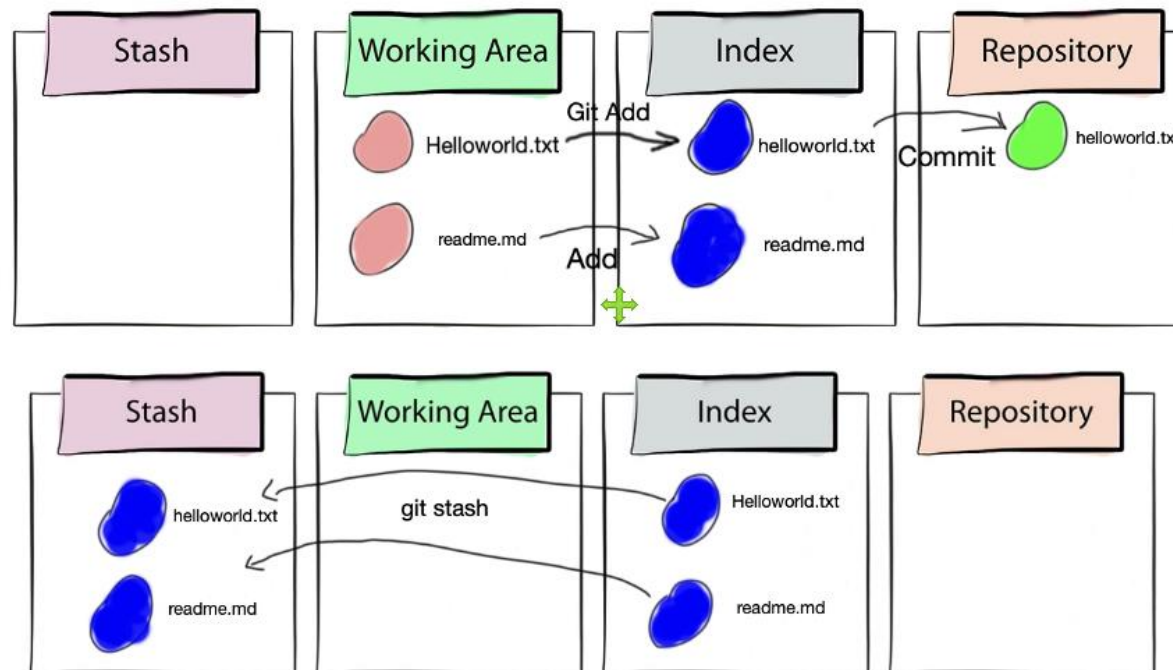
# 01

## 커밋 되돌리기 git reset



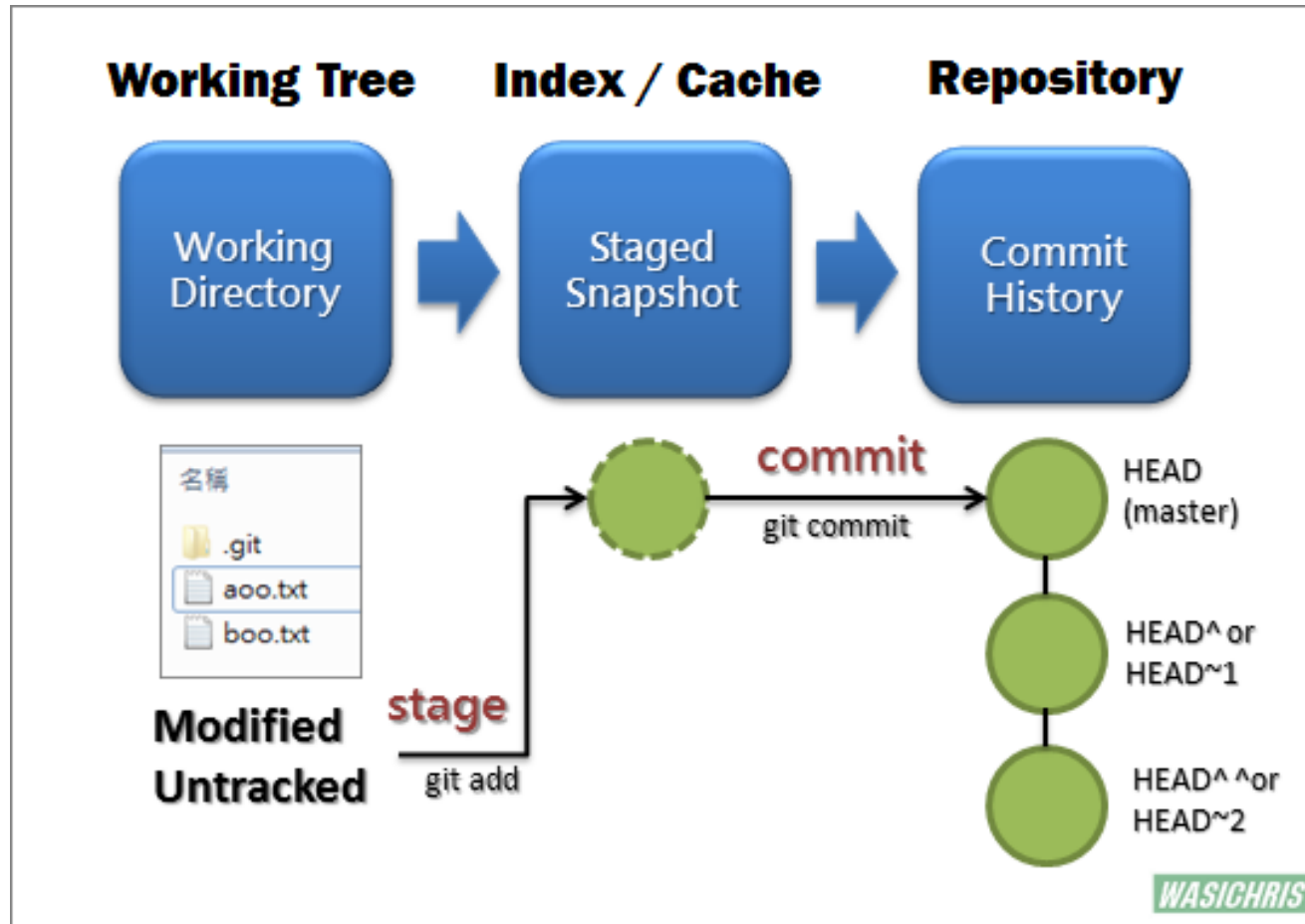
# Git의 3+1 = 4가지 영역

- Working directory, 작업공간, 작업트리, 작업디렉토리
- Stage area, 스테이지 영역, 인덱스(index), 캐쉬(cache)
- Git repository



영역	역할	위치
Working Directory	프로젝트 디렉토리이며, 개발자가 직접 코드를 수정하는 공간을 의미합니다.	.
Index (Staging Area)	Working Directory 에서 Repository로 정보가 저장되기 전 준비 영역입니다.	.git/index
Repository	파일이나 폴더를 변경 이력별로 저장해두는 곳입니다.	.git
Stash	임시적으로 작업사항을 저장해두고, 나중에 꺼내올 수 있는 영역입니다.	.git/refs/stash

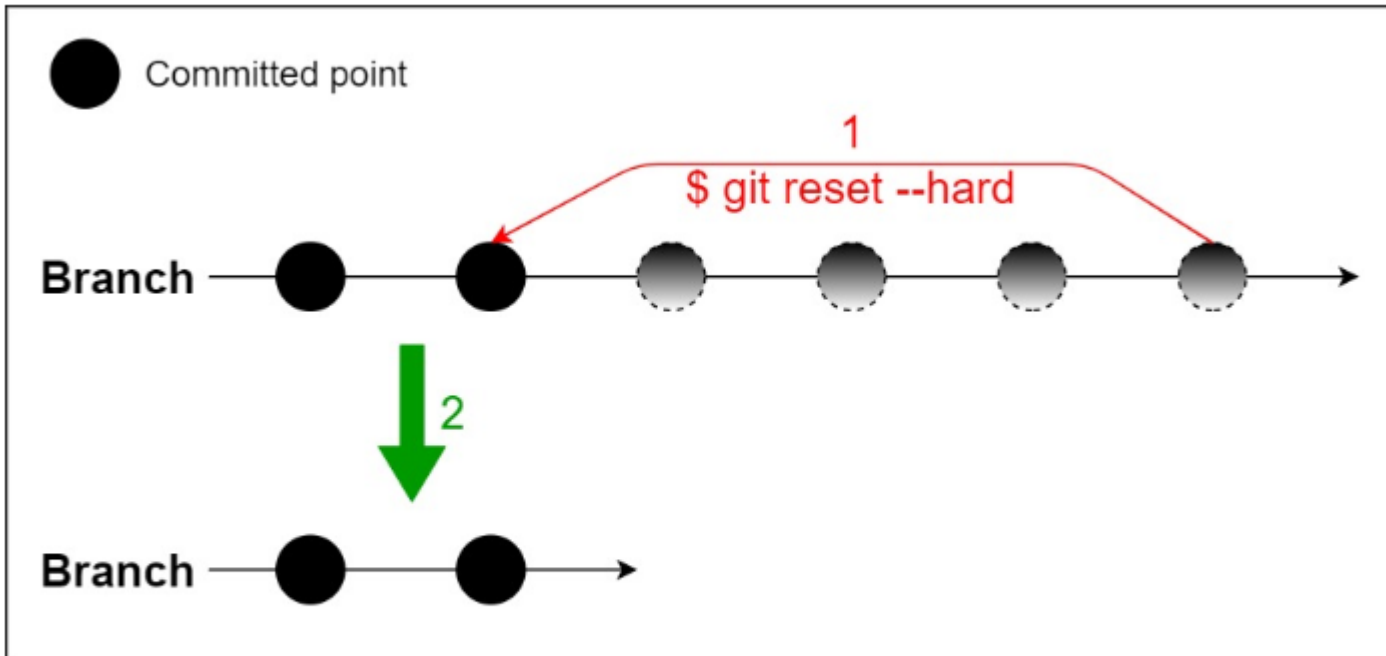
- **1. Working Directory(작업영역)**
  - 프로젝트 디렉토리이며, 개발자가 직접 코드를 수정하는 공간을 의미
  - .git을 제외한 모든 영역에 해당
- **2. Index(Staging Area), cache**
  - Working Directory 에서 Repository로 정보가 저장되기 전 준비 영역
  - .git/index 파일로 관리
- **3. Repository(저장소)**
  - 파일이나 폴더를 변경 이력 별로 저장해두는 영역
  - .git 디렉토리 내에 존재
  - Local, Remote Repository로 구분
- **4. Stash(임시영역)**
  - 임시적으로 작업 사항을 저장해두고, 나중에 꺼내올 수 있는 영역



<https://dotblogs.com.tw/wasichris/2016/04/29/225157>

- 특정 커밋으로 되돌아 가기(Roll back)

- 이전으로 돌아가는 커밋 이후 이력(history)이 모두 삭제
  - 돌아가려는 커밋으로 저장소는 재설정되고, 해당 커밋 이후의 이력은 제거
- Reset은 시계를 다시 뒤로 맞추는 것으로 '타임 머신'과도 같음
- 가능하면 자신의 저장소에서만 사용하도록



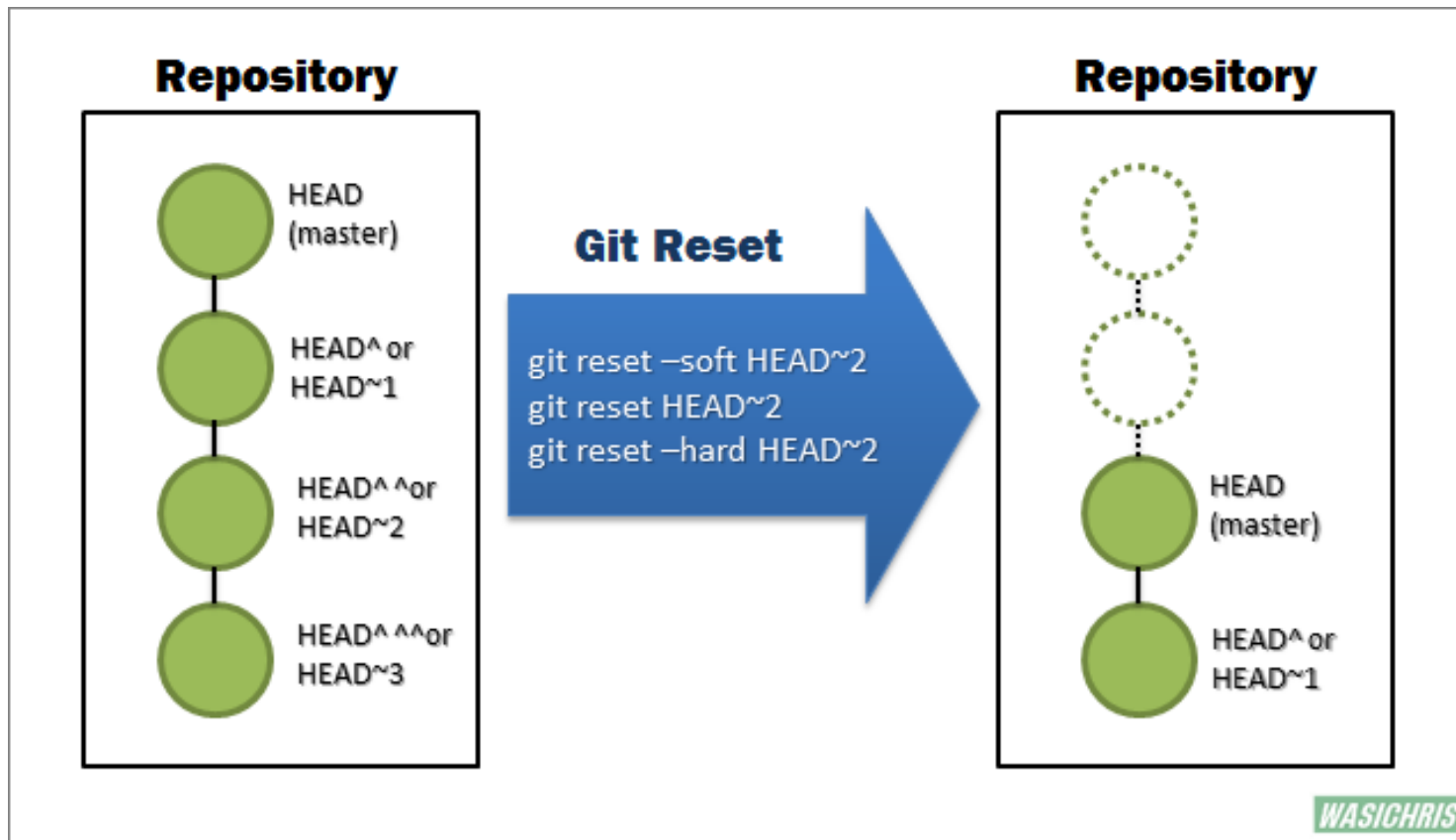
Bash Copy

<https://gogojuni.dev/git-reset/>

# Reset의 옵션 3가지

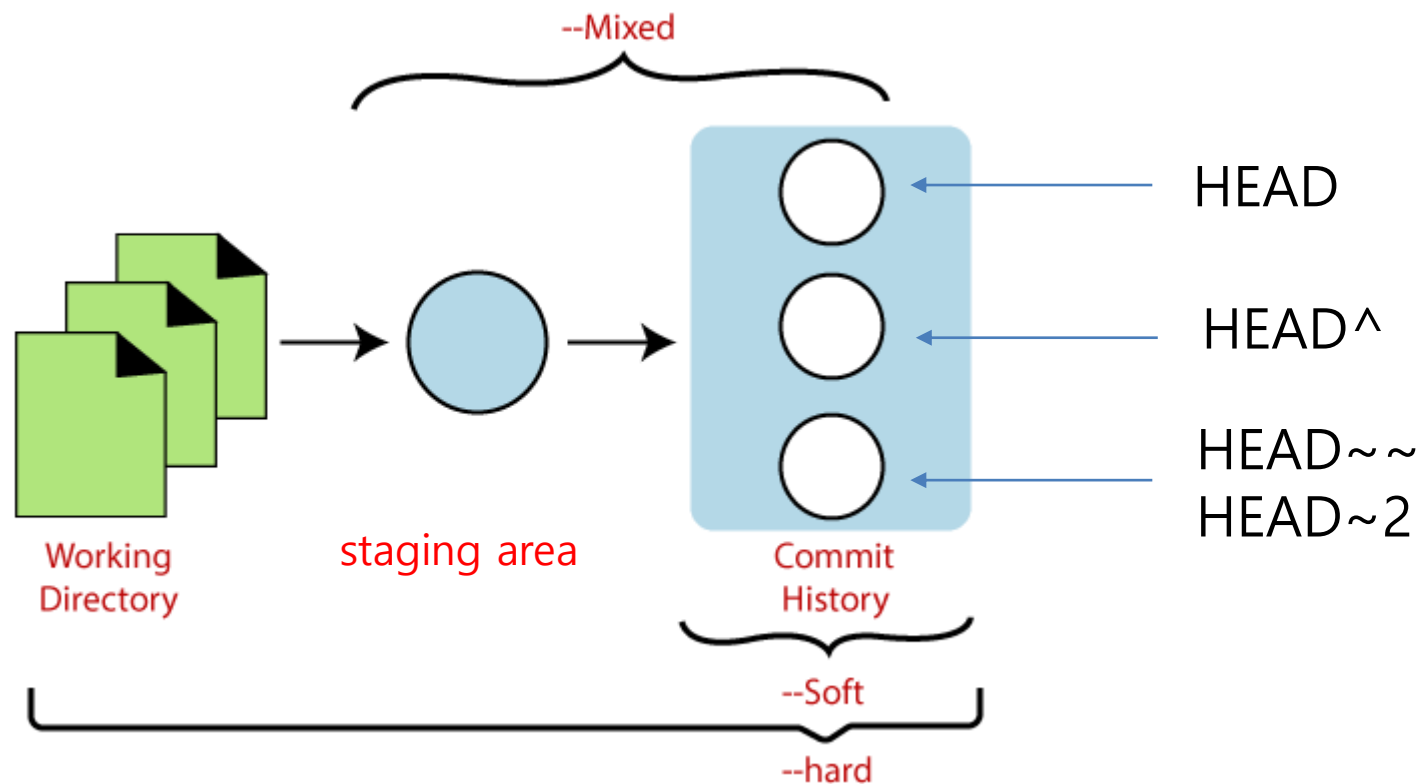
- \$ git reset --soft HEAD~3
- \$ git reset --mixed HEAD~3
  - \$ git reset HEAD~3
- \$ git reset --hard HEAD~3

```
git reset --soft  
git reset --mixed  
git reset --hard
```





- \$ git reset



- **--soft**

- Reset 이후에 커밋 내용만 수정되고  
작업공간(WD)과 index(stage)는  
이전과 같음

- `$ git reset --soft 9b45f`
- `$ git reset --soft HEAD^`

- **--mixed(없거나)**

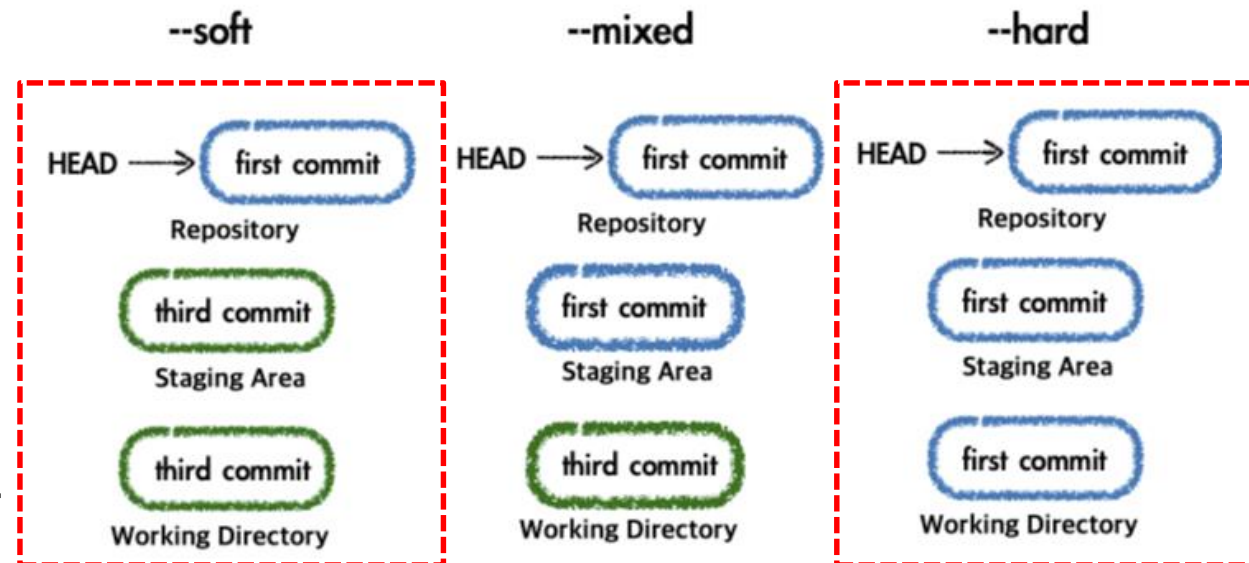
- Reset 이후에 커밋 내용과  
index(stage)가 같음

- `$ git reset 9b45f`
- `$ git reset HEAD^`
- `$ git reset --mixed HEAD^`

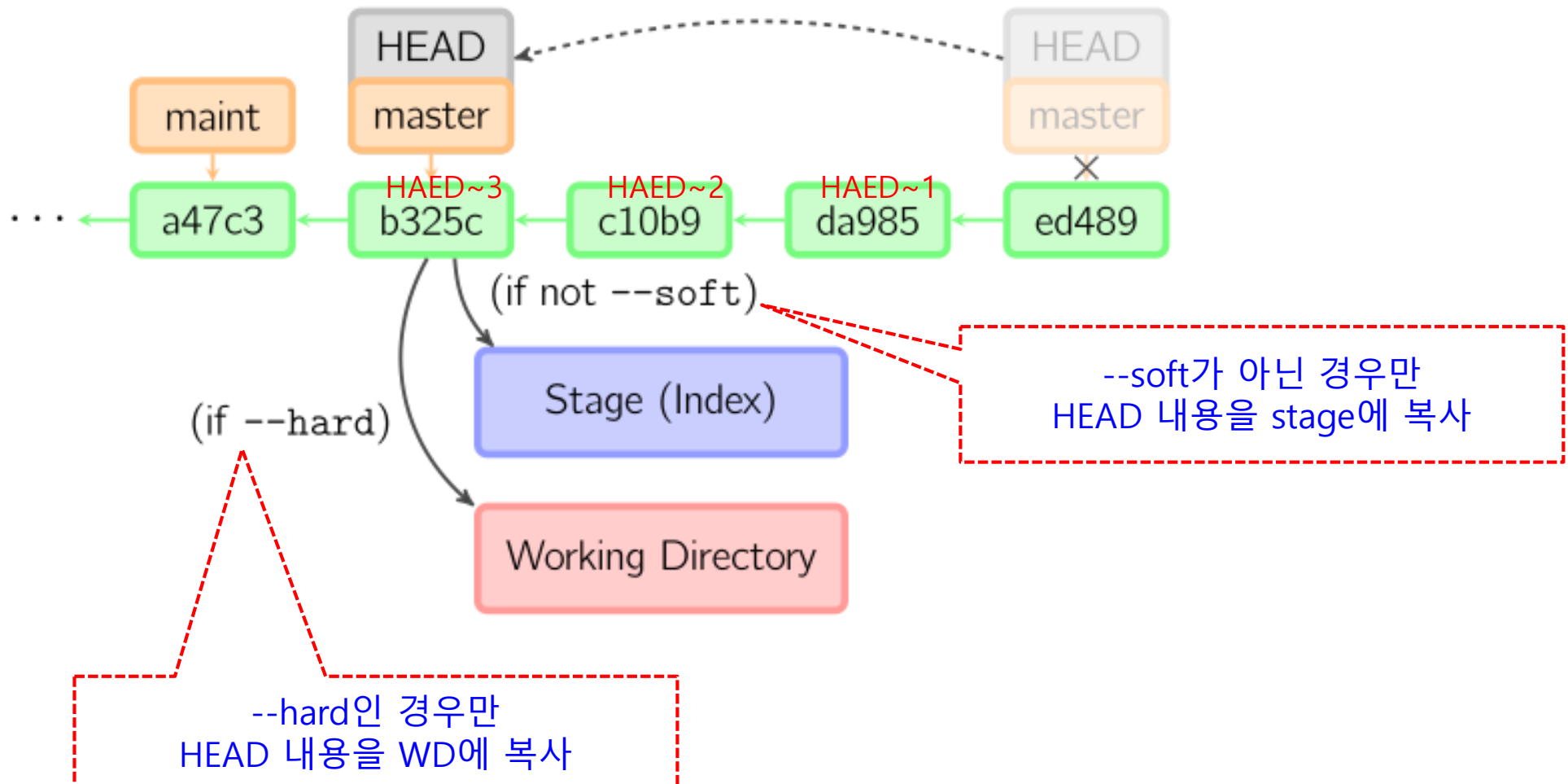
- **--hard:**

- Reset 이후에 커밋 내용으로 작업공간  
(WD)과 index(stage) 모두 동일

- `$ git reset --hard 9b45f`
- `$ git reset --hard HEAD^`
- `$ git reset --hard HEAD~~`



git reset HEAD~3



# \$ git reset --soft HEAD~

깃과 깃허브 Python language

## • 이전



## • 이후

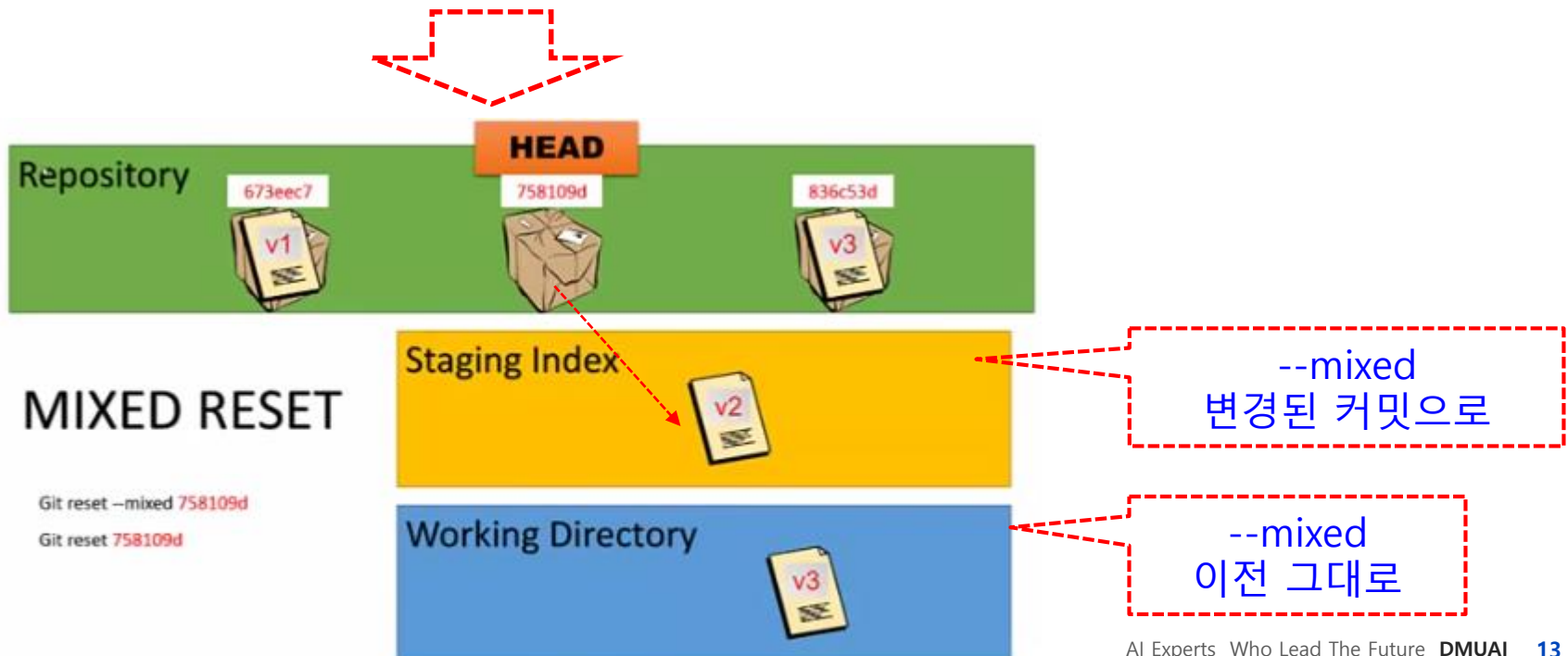


# \$ git reset --mixed HEAD~

- 이전



- 이후

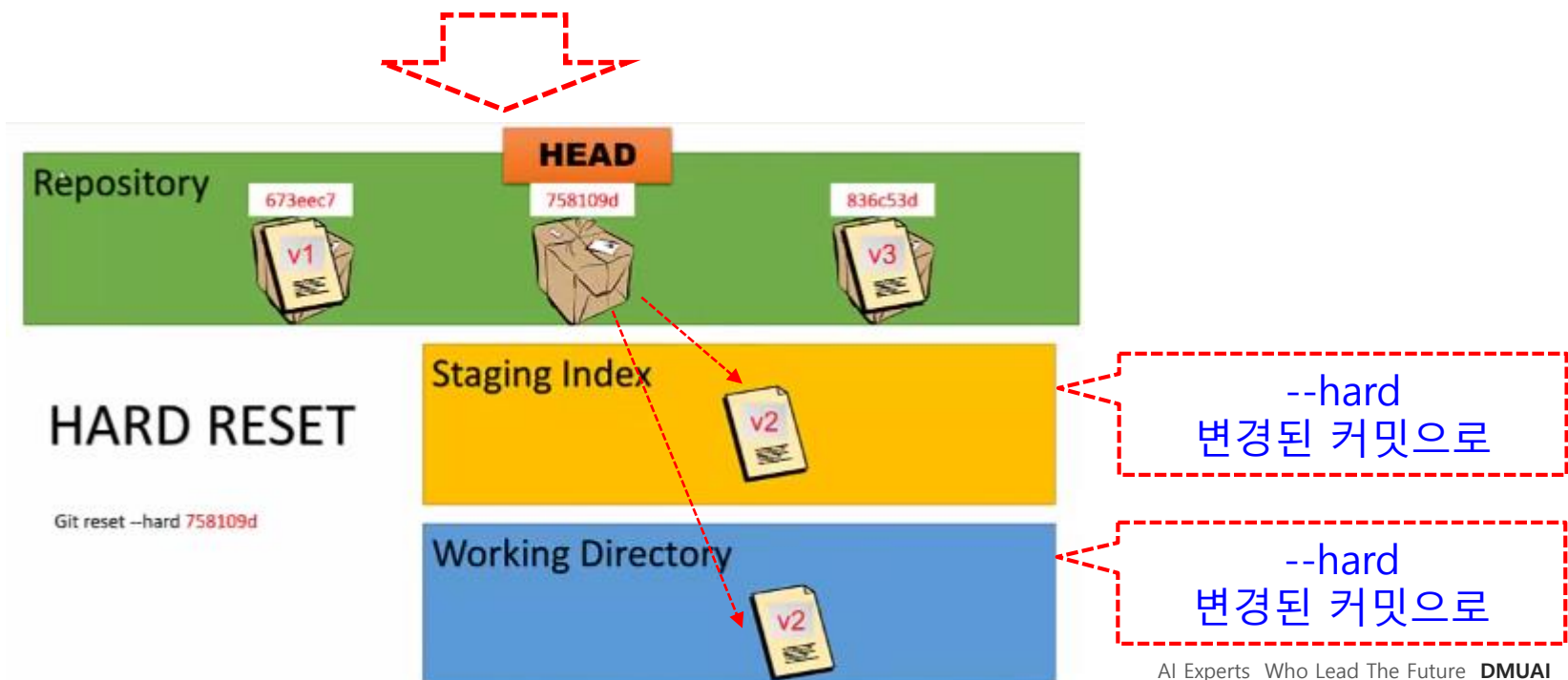


# \$ git reset --hard HEAD~

- 이전



- 이후



- **\$ 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
- **Reset 이후에 다시 이전 상태로 돌아가려면**
  - 다음을 특별히 제공
    - **\$ git reset --hard ORIG\_HEAD**

AI Experts  
Who Lead  
The Future

## 02

이전 로그를 남기고 새로운 커밋으로  
커밋 되돌리기 `git revert`

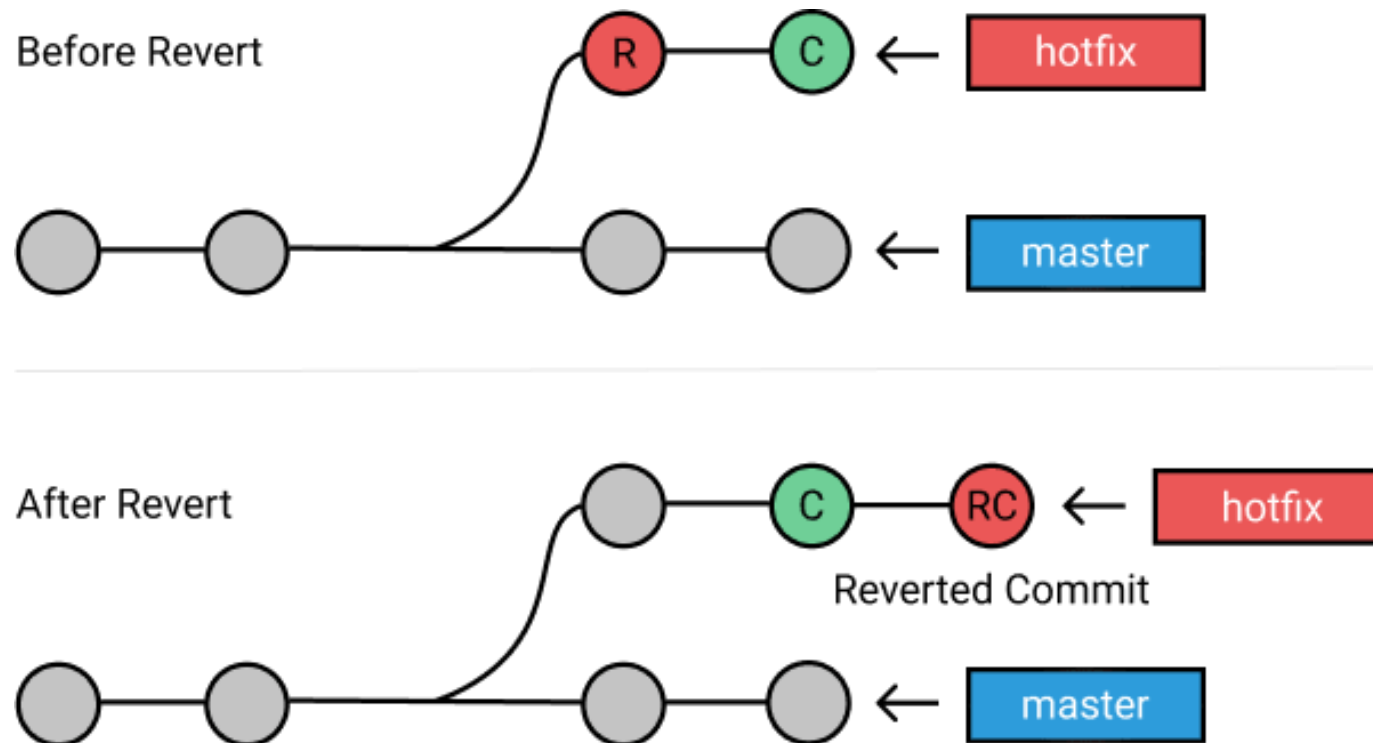




- 취소, undo 수행

- 지정한 커밋을 취소해 바로 이전 상태로 되돌리는 방법

- 커밋해온 모든 변경 사항을 rollback
- 이전의 커밋 히스토리는 그대로 유지되며, 이것을 되돌리는 새로운 커밋이 그 이후에 추가



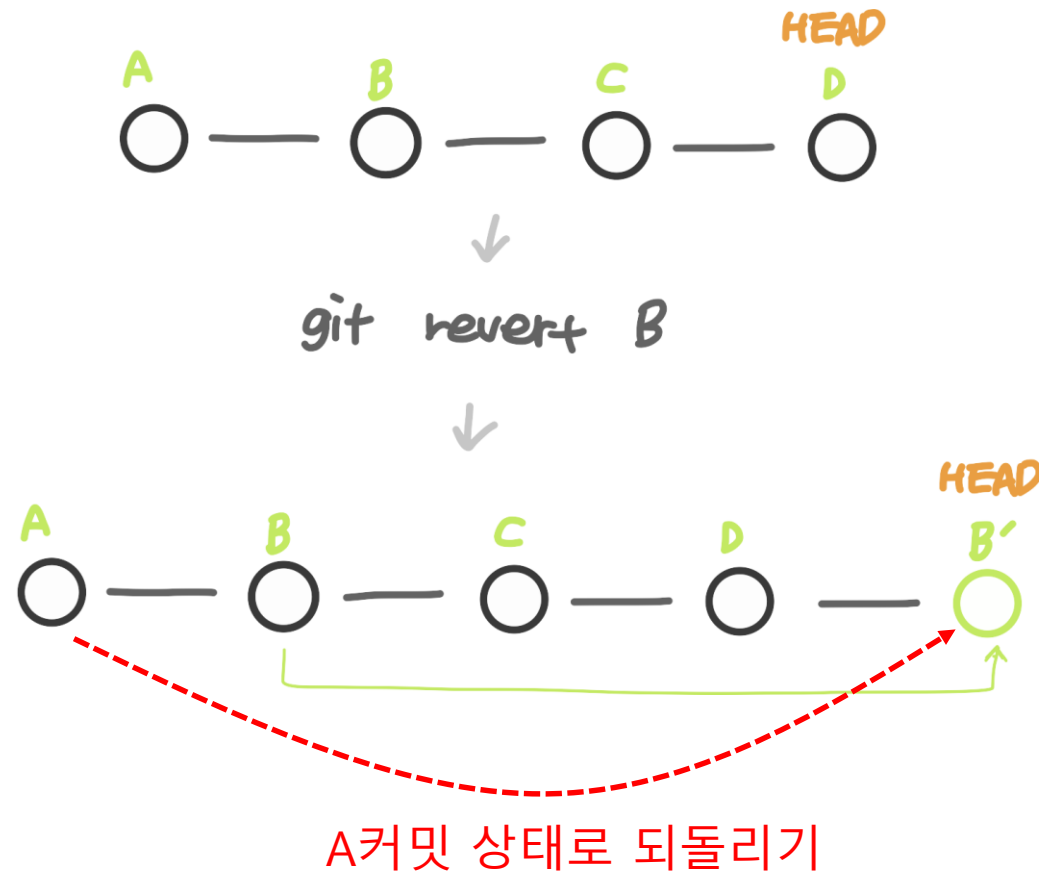
- `$ git revert <commitid>`

- <commitid> 바로 이전으로 이동

- commit을 추가로 수행
    - 커밋 메시지 입력 편집기 실행
      - 기본 메시지: 'revert 이전\_메시지'

- 과거의 모든 커밋은 그대로 유지 관리

- 새로운 커밋을 추가해 특정 시점 <commitid> 이전으로 이동



## • Revert 이전



## • \$ git revert HEAD

- [master d47bb1d] Revert "pull의 설명을 추가"
- 1 files changed, 1 insertions(+), 2 deletions(-)

기본 편집기에서  
커밋메시지 편집 화면 실행

## • \$ git revert HEAD --no-edit

- [master d47bb1d] Revert "pull의 설명을 추가"
- 1 files changed, 1 insertions(+), 2 deletions(-)

편집 화면 실행없이  
자동 커밋 메시지로 커밋  
Revert "이전 메시지"

## • Revert 이후



AI Experts  
Who Lead  
The Future

## 03

### git reset vs revert 비교

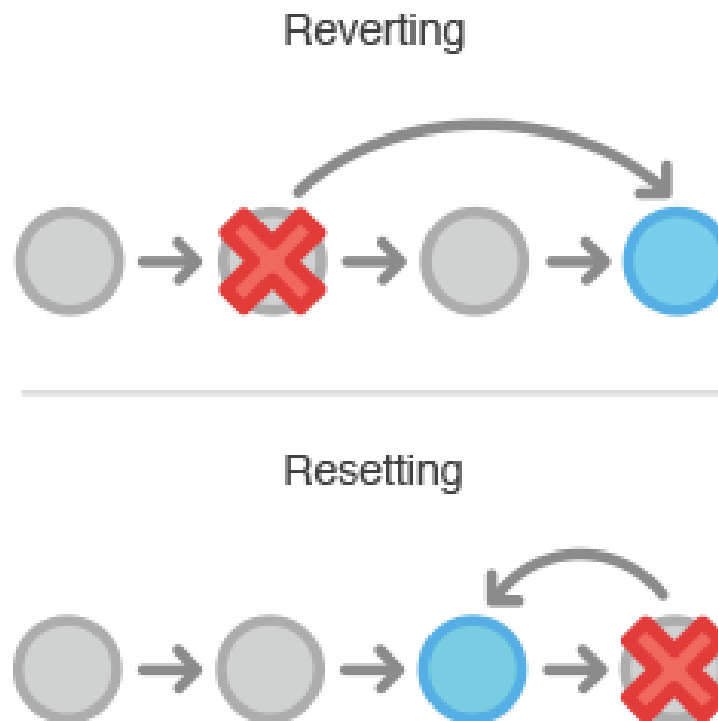


- 옵션에 따른 차이

	working directory working tree working copy	index staging area cache	repository history tree
			git reset --soft
		git reset --mixed	
	git reset --hard		

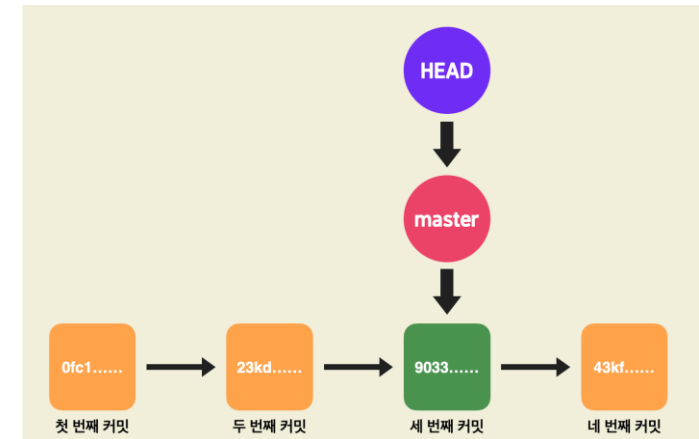
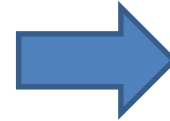
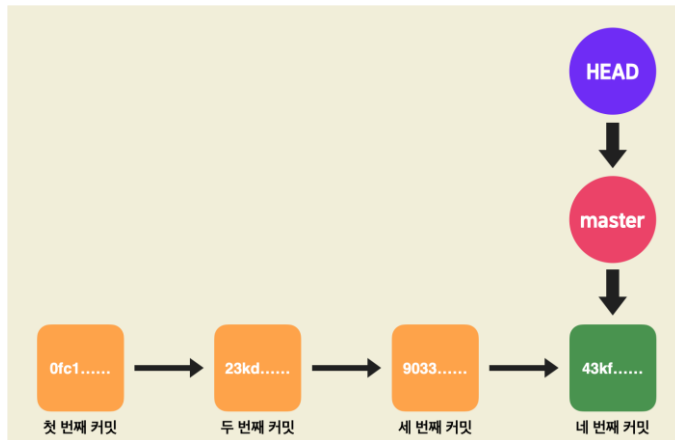
Git reset의 3가지 옵션			
	--soft	--mixed(default)	--hard
HEAD	지정한 commit 으로 이동한다.		
HEAD가 가리키는 Branch	HEAD와 같이 움직인다.		
Staging Area	변화 X	지정한 commit과 동일 내용	지정한 commit과 동일 내용
Working Directory	변화 X	변화 X	지정한 commit과 동일 내용
주 용도	branch 이동하기	Staging Area에서 빼기	commit 되돌리기

- 재지정과 취소



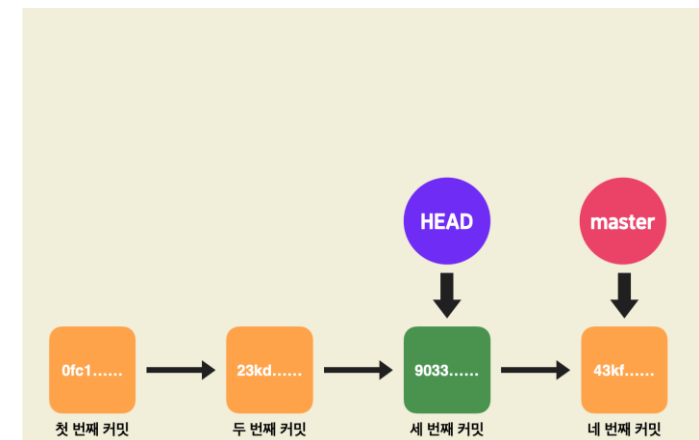
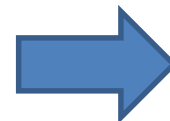
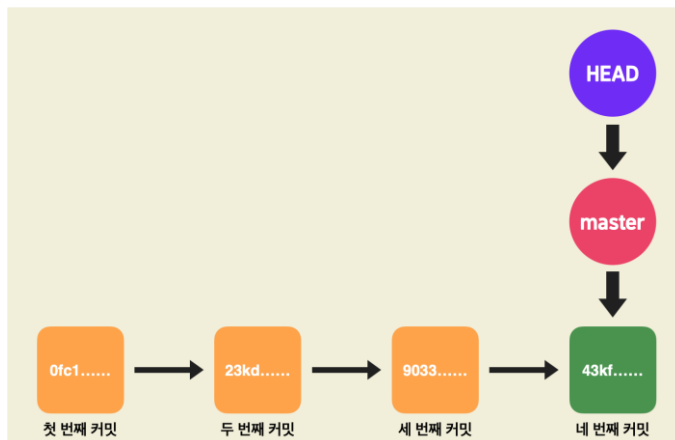
- **git reset 9033**

- 일반적으로 다시 앞으로 가기가 어려움



- **git checkout 9033**

- 바로 checkout으로 다시 앞으로 가기가 수월



- 'ORIG\_HEAD'
  - 바로 이전 커밋 참조
    - `$ git reset --hard ORIG_HEAD`



## 실습 후 우리는 다음 ~~~~~할 수 있다.

깃과 깃허브 Python language

- 커밋 이력을 되돌리는 방법을 이해하고 수행할 수 있다.
  - 커밋 이력 재설정(reset)을 수행할 수 있다.
  - 특정 커밋 이력 취소하고 바로 이전 상태로 돌아가는 방법(revert)을 수행할 수 있다.
- 특히 Reset vs Revert의 차이를 이해하고 활용할 수 있다.
  - Reset은 되돌리는 이후 커밋 이력이 제거되고
  - Revert는 현재까지의 커밋 이력에 되돌리기 커밋이 추가됨