

# 오픈소스sw의 이해

## Lecture #4: Version management II

Software Engineering laboratory  
Yeungnam University



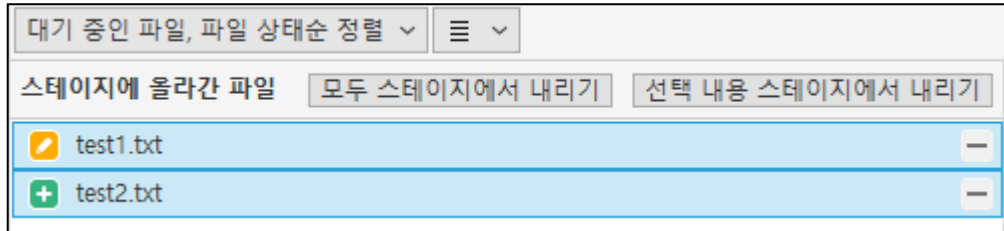
# 작업 되돌리기

---

- 하나의 버전을 생성하기 위한 과정 (요약)
  - Working directory에서 변경 사항 발생
  - Stage로 올리기
  - commit 하기
- 변경된 파일들을 취소하려면?
  - stage에 올라간 파일은 어떻게 취소해야 할까?
  - 아직 stage로 올리지 않은 변경된 파일을 취소하려면 어떻게 해야 할까?
  - 이미 commit한 파일은 어떻게 취소할 수 있을까?

# 작업 되돌리기

- stage에 올라간 파일 되돌리기
  - “선택 내용 스테이지에서 내리기” 클릭

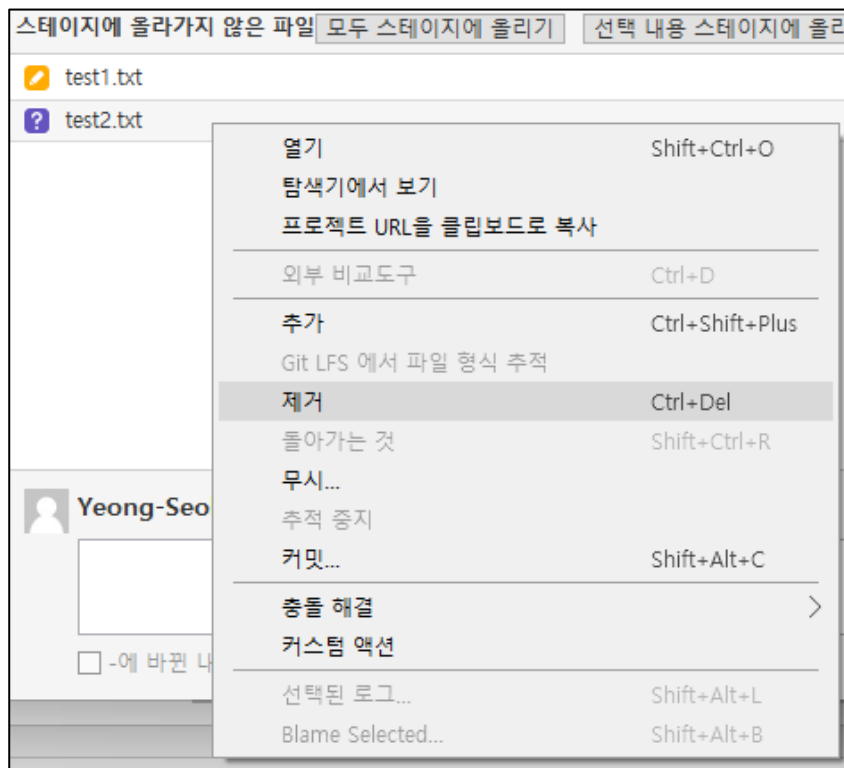


# 작업 되돌리기

## ■ stage에 올라가지 않은 파일 되돌리기

- 돌아가는 것: 기존에 존재하고 있던 파일에 대한 수정 취소. 즉 변경사항을 폐기.
- 제거: 새롭게 생성된 파일의 변경 사항을 취소. 즉 파일이 만들어지기 전으로 돌아간다?

삭제의 의미 *한번이상 커밋되었을 때 (4점)*



# 작업 되돌리기

---

## ■ commit 되돌리기

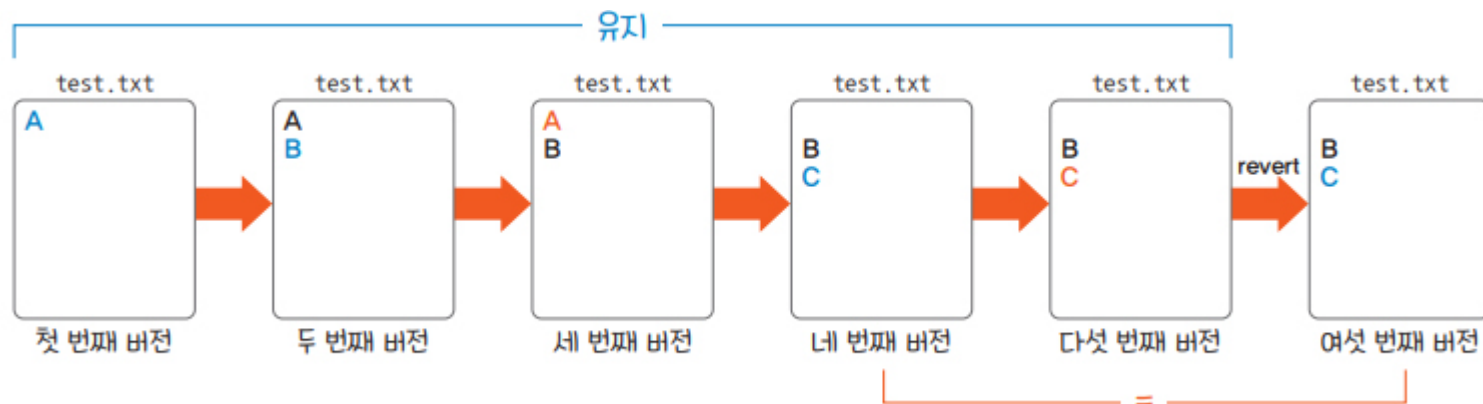
- 이전 돌아가고 싶은 커밋으로 가고 싶을 때 사용
  - 릴리즈를 했는데 치명적인 버그로 인하여 롤백해야하는 경우 이전 커밋으로 되돌아가기 위해 사용
- revert vs. reset

# 작업 되돌리기

## ■ commit한 파일 취소하는 방법 2가지

### ● revert

- 버전을 되돌리되, 되돌아간 상태에 대한 새로운 버전(커밋)을 만드는 방식.
- 중요한 점은 기존의 버전은 삭제되지 않는다는 점.
- (예) 다섯 번째 버전을 revert하면 다음 그림과 같이 네 번째 버전으로 되돌아간 새로운 여섯 번째 커밋이 만들어짐
  - 첫 번째 버전부터 다섯 번째 버전은 그대로 유지됨.



# 작업 되돌리기

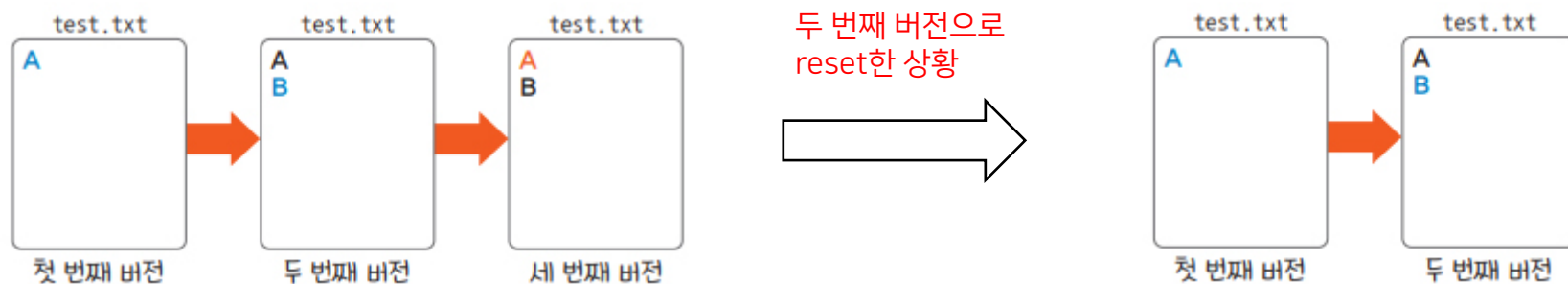
## ■ commit한 파일 취소하는 방법 2가지

### ● reset

- 되돌아갈 버전의 시점으로 완전하게 되돌아가는 방식.

- 즉, 되돌아갈 버전 이후의 버전은 삭제되는 방식.

- reset의 옵션 3가지: soft, mixed, hard



# 작업 되돌리기

## ■ commit한 파일 취소하는 방법 2가지

### ● reset --soft

- 작업 디렉터리 내 변경 사항과 스테이지에 추가된 변경 사항은 유지, 커밋했다는 사실만 되돌리는 reset.
- 변경 이력은 모두 삭제하지만 변경 내용은 남아있음. stage 되어있음.





# 작업 되돌리기

## ■ commit한 파일 취소하는 방법 2가지

### ● reset --mixed

- 작업 디렉터리 내 변경 사항은 유지하되, 스테이지와 커밋을 되돌리는 reset.
- 변경 이력은 모두 삭제하지만 변경 내용은 남아있음. unstage 상태로 코드는 남아 있음.



# 작업 되돌리기

## ■ commit한 파일 취소하는 방법 2가지

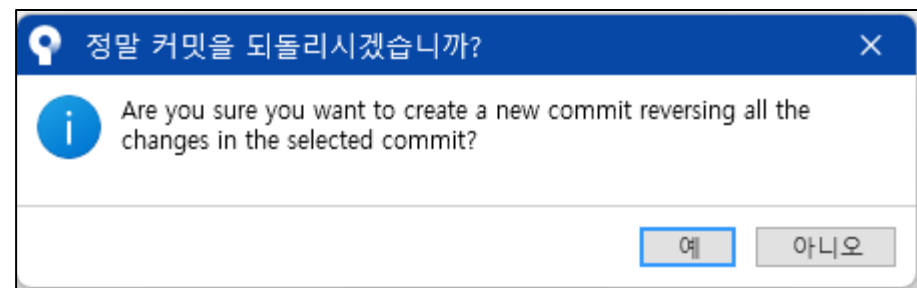
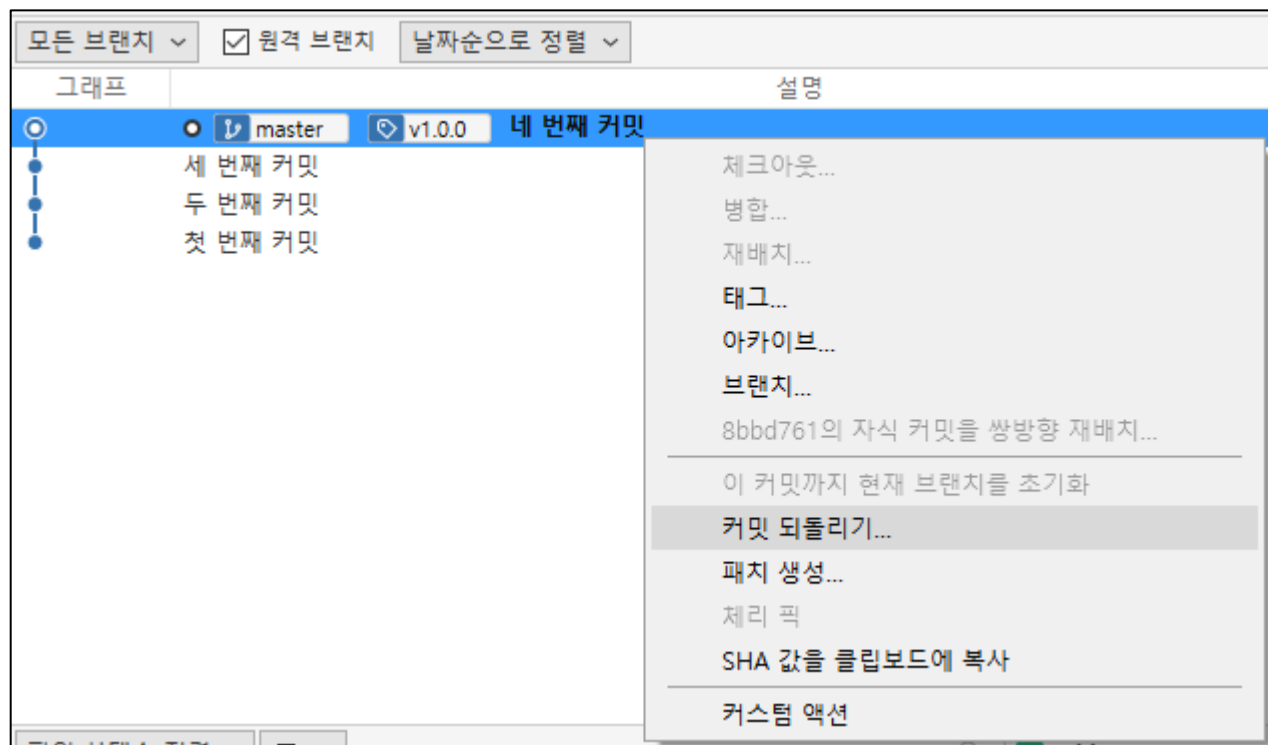
### ● reset --hard

- 마지막으로 작업 디렉터리 내 변경 사항까지 통째로 되돌리는 reset.
- 돌아간 커밋 이후의 변경 이력은 모두 삭제.



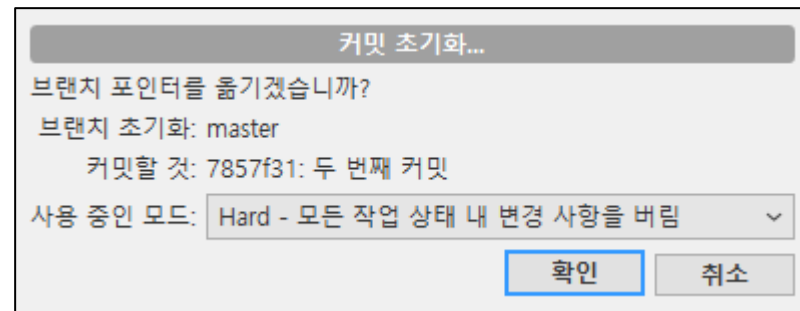
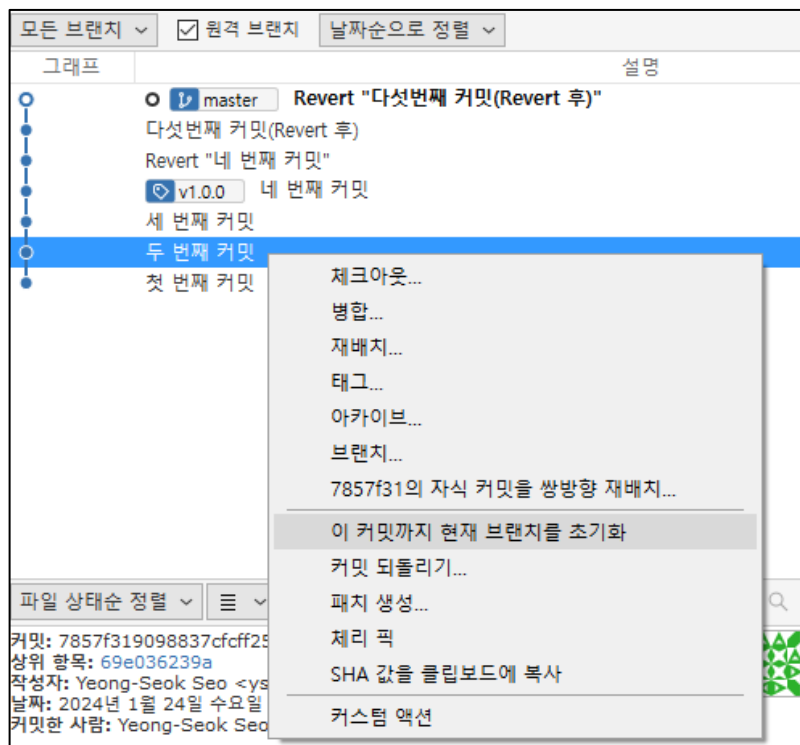
# (실습) revert in SourceTree

- revert 하기 위한 커밋에서 마우스 우측버튼
  - “커밋 되돌리기” 클릭.



# (실습) reset in SourceTree

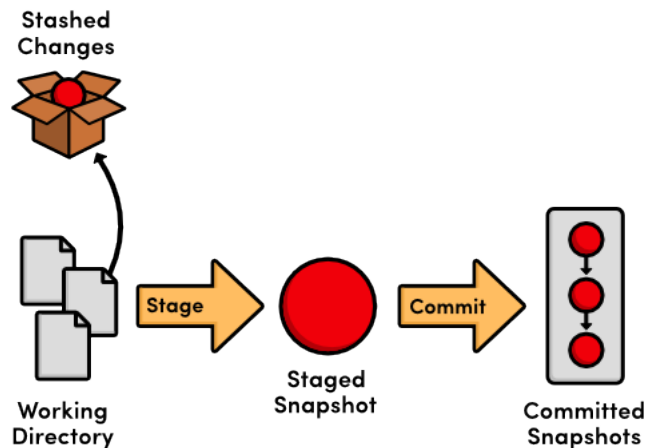
- reset 하기 위한 커밋에서 마우스 우측버튼
  - “이 커밋까지 현재 브랜치를 초기화” 클릭.
  - soft나 mixed 옵션일 경우, 해당 커밋 이후에 변경된 파일들은 “커밋하지 않은 변경사항”이라는 항목의 브랜치로 생성하여 (staging or unstaging 상태로) 유지함.



# 작업 임시 저장하기

## ■ stash(스태시)

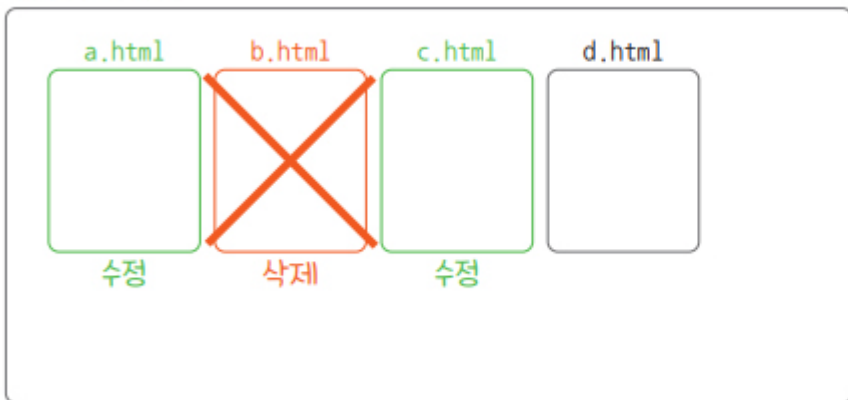
- 아직 마무리하지 않은 작업을 스택에 잠시 저장할 수 있도록 하는 명령어
  - 아직 완료하지 않은 일을 commit하지 않고 나중에 다시 가져와 마무리할 수 있음.
  - 자신이 어떤 작업을 하던 중에 다른 요청이 들어와 하던 작업을 멈추고 잠시 브랜치를 변경해야 할 일이 있을 경우, 아직 완료하지 않은 일을 commit하는 것은 껄끄럽기 때문에 stash 이용.
- 아래에 해당하는 파일들을 보관해두는 장소
  - Modified이면서 Tracked 상태인 파일 (tracked: 과거에 이미 commit했던 관리 대상 상태의 파일)
  - Staging area에 있는 파일 (staged 상태의 파일)  
새롭게 생성된 파일 X



# 작업 임시 저장하기

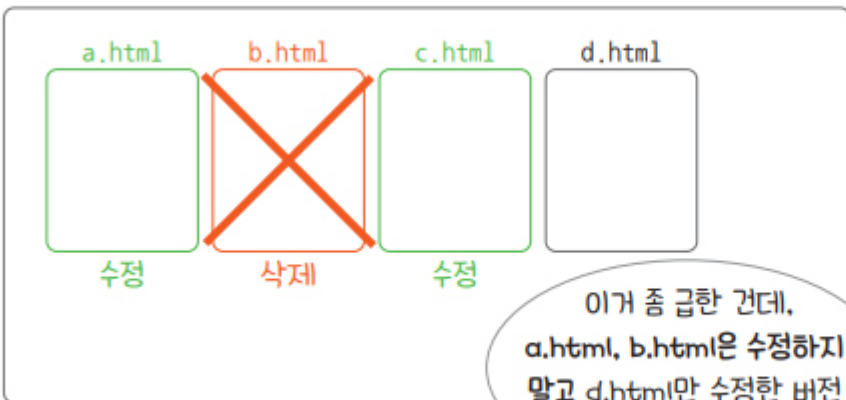
## ■ stash(스태시) 적용 상황

작업 디렉터리



이렇게 변경하긴 했는데,  
버리긴 아깝고  
쓰자니 아쉽단 말이지.

작업 디렉터리



이거 좀 급한 건데,  
a.html, b.html은 수정하지  
말고 d.html만 수정한 버전  
좀 만들어줘요.



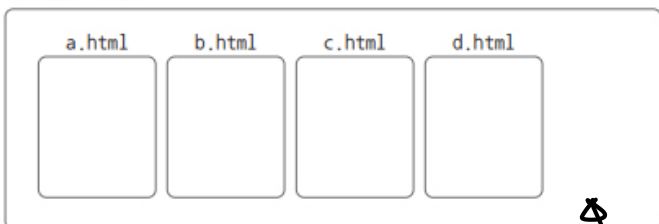
헉, 지금까지  
이렇게 변경했는데?

# 작업 임시 저장하기

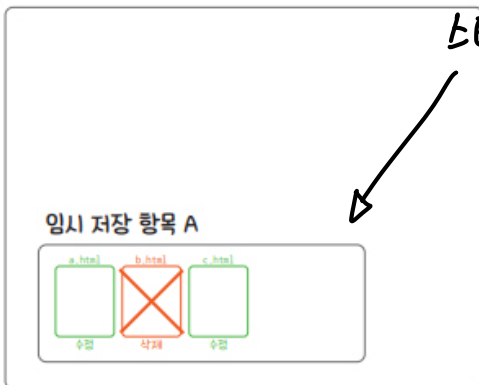
## ■ stash(스태시)로 작업 내역 임시 저장하기

- 스테시를 하게 되면 작업 디렉토리에 생성한 모든 변경 사항이 임시 저장됨.
  - 다수의 임시 저장 항목 가능.
- 그 후 작업 디렉토리는 변경 사항이 생기기 전의 상태로 돌아감.

작업 디렉터리

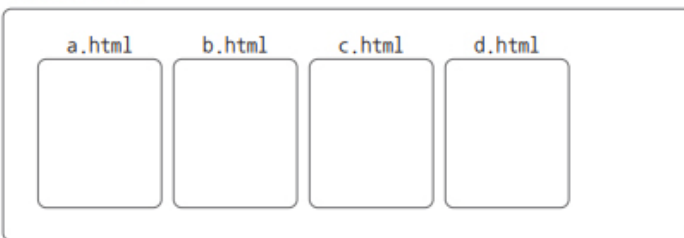


스태시



스테시 후 기존 버전로 돌아감

작업 디렉터리



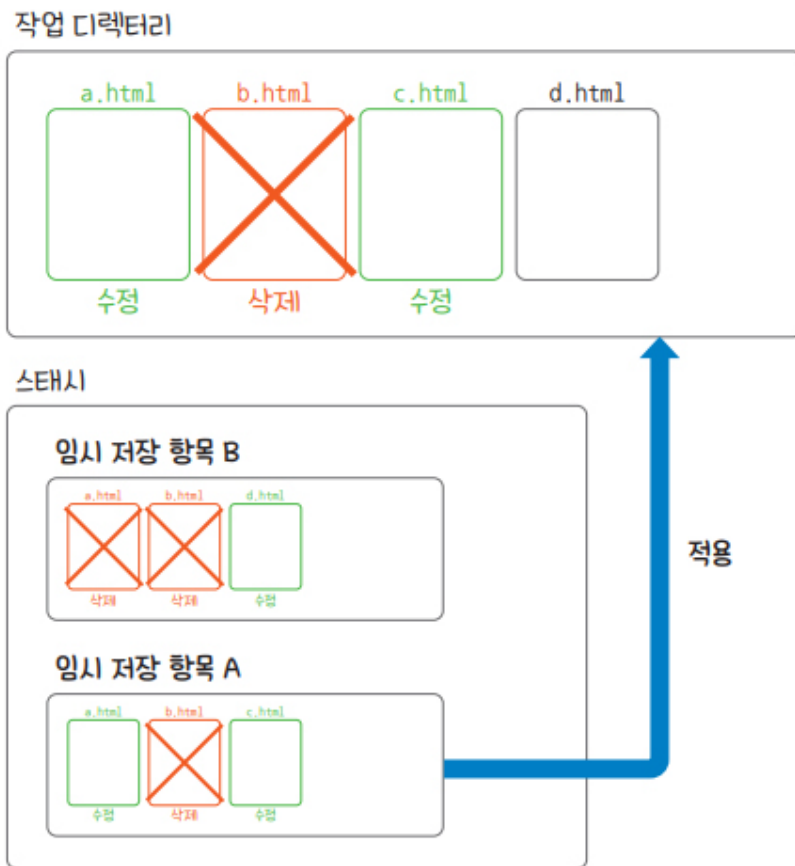
스태시



# 작업 임시 저장하기

## ■ stash(스태시) 로 임시 저장된 변경사항 적용하기

- 스테시로 임시 저장된 변경 사항들은 언제든지 다시 꺼내어 작업 디렉터리에 다시 적용할 수 있음.





# 작업 임시 저장하기

---

## ■ stash(스태시) 적용시 주의사항

- 스테시는 git이 변경 사항을 추적하는(tracked) 파일에만 사용할 수 있음.
  - 스테이지에 이미 올라와 있거나 한번이라도 커밋한 적이 있는 파일에만 사용할 수 있음.
- 방금 막 생성한 파일처럼 git이 기존에 변경 사항을 추적하지 않은(untracked) 파일에는 스테시를 사용할 수 없음.

# (실습) stash in SourceTree

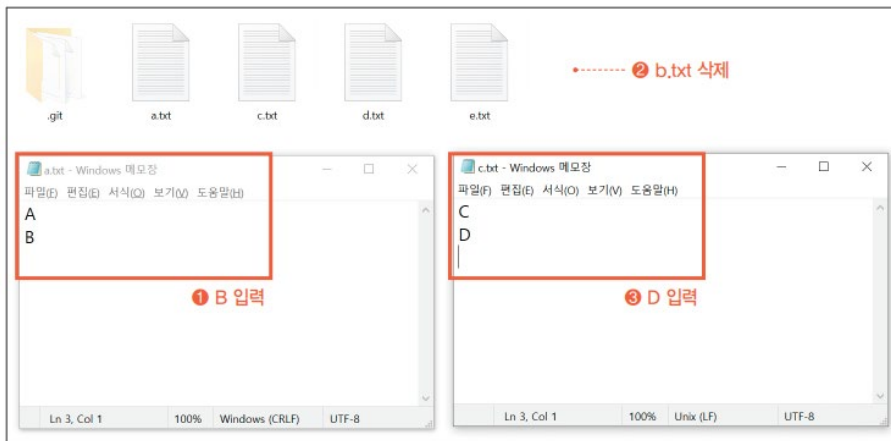
## ■ 스테시로 작업 임시 저장하기

### ● 파일 준비

- 임의의 로컬 저장소에 a.txt, b.txt, c.txt, d.txt, e.txt 파일을 만들고 이 다섯 개의 텍스트 파일에 각각 A, B, C, D, E를 저장 후, staging 및 commit 하기

### ● 변경 사항 생성

- a.txt 파일 안에 B를 추가하고, b.txt 파일은 삭제
- c.txt 파일 안에 D를 추가

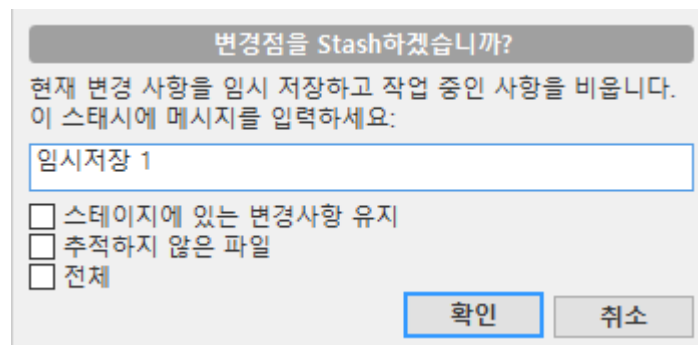
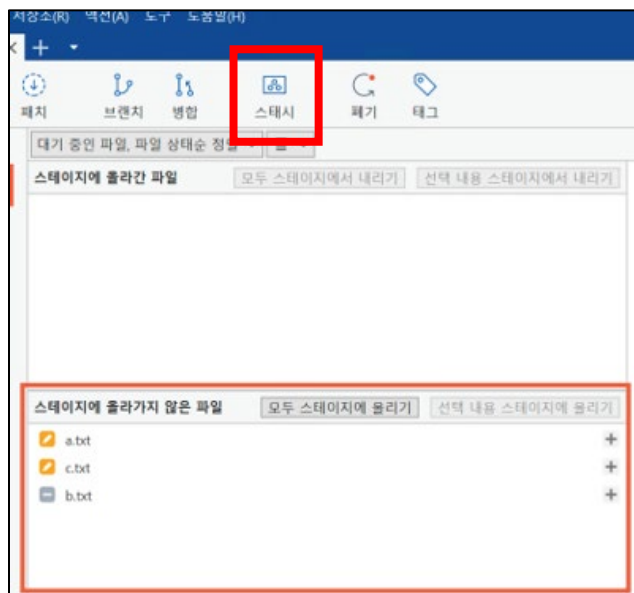


# (실습) stash in SourceTree

## ■ 스테시로 작업 임시 저장하기

### ● 작업 내역 임시 저장

- 상단의 “스테시”를 클릭
  - ‘변경점을 Stash하겠습니까?’라는 물음에 대해 ‘임시저장 1’을 입력
  - 스테이지에 있는 변경사항 유지 항목은 체크하지 않고 확인을 클릭하면 좌측 스테시 메뉴에 임시저장 생성.
  - 탐색기에서 작업 파일들이 원래대로 복구됨 확인.



# (실습) stash in SourceTree

---

## ■ 스테시로 작업 임시 저장하기

- 추가 변경 사항 생성

- 이번에는 d.txt 파일과 e.txt 파일을 삭제해보기

- stash 하기

- 위 두 개의 파일을 삭제한 변경 사항을 또 다시 임시 저장하기
- 스테시 메시지에는 '임시저장 2'를 적고, 확인 버튼 클릭

# (실습) stash in SourceTree

## ■ 임시 저장된 작업 내역 확인

- 스테시에 임시저장1, 임시저장2 클릭해보기
- 스테시 적용하기 위해서는 마우스 우측버튼 클릭 후 스테시 적용 클릭
  - 파일 상태 체크해보면 원래 상태로 복구된 것을 확인 가능함.

