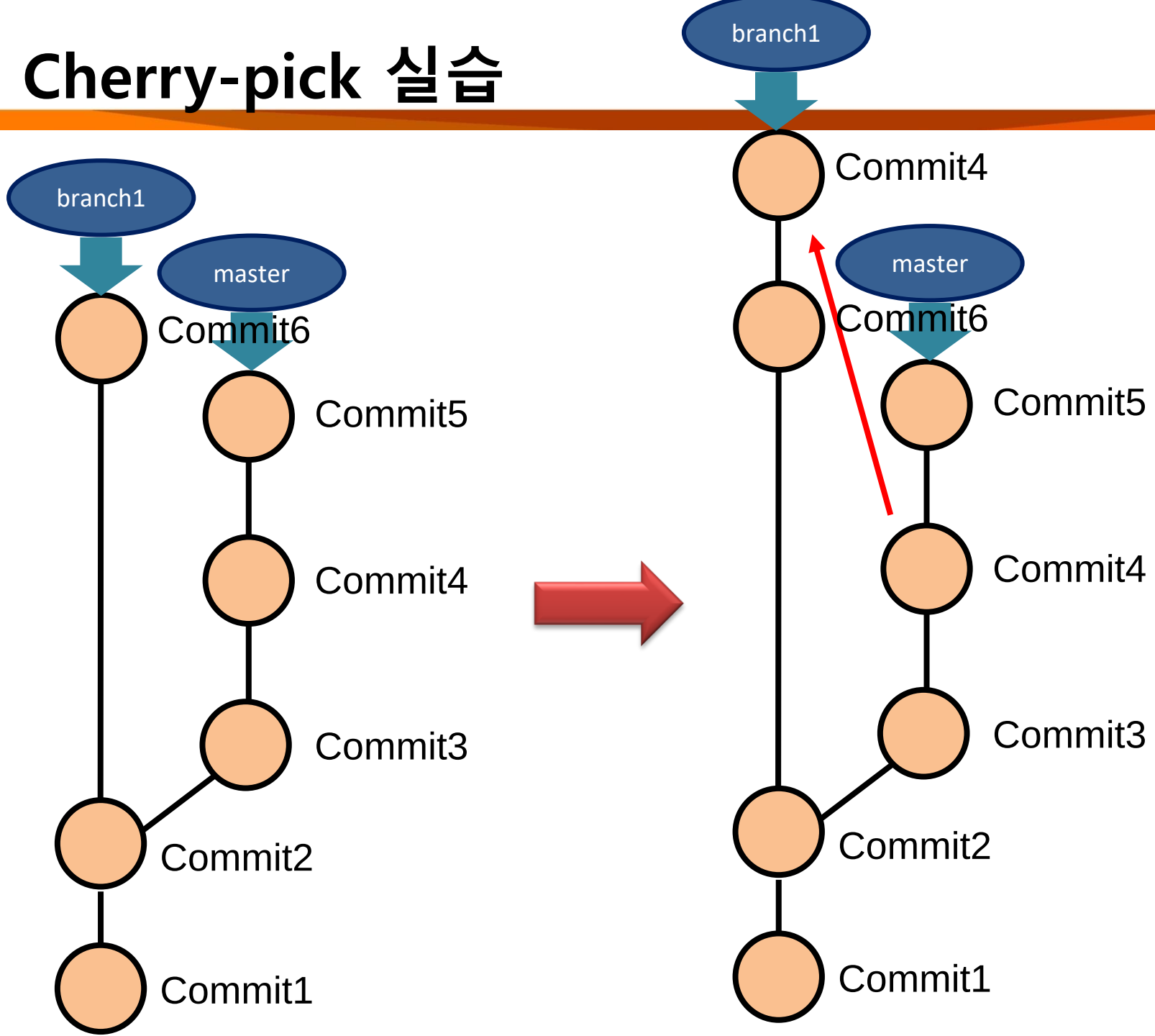
GUI 환경에서의 cherry-pick, reset, revert, stash

세종대학교 이은상

Cherry-pick

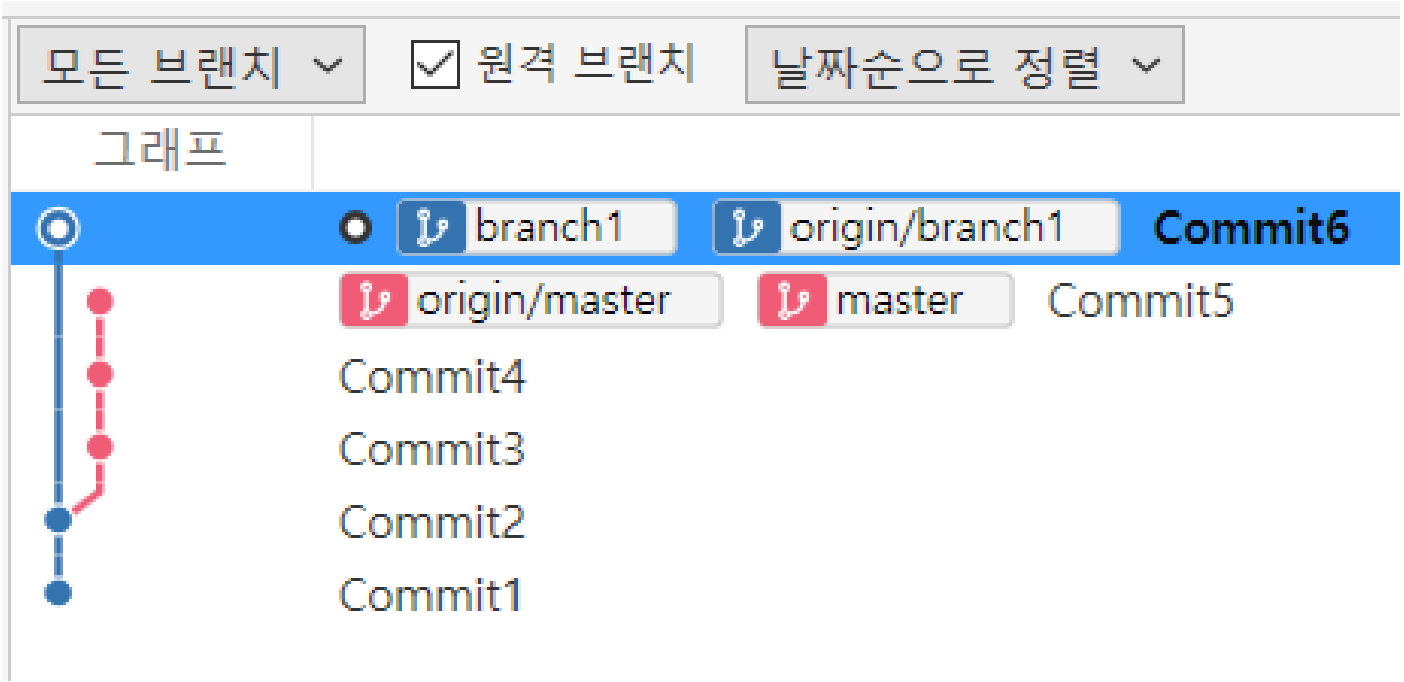
- latest 브랜치에 코드를 막 출시한 후에 당장 고쳐야하는 버그가 있는 것을 알게 됨
- 이후 master 브랜치에 고친 버그가 커밋되었는데 master 브랜치의 다른 변경사항은 말고 딱 그 버그 고친 커밋만 반영하려고 하는 상황
- 다른 기능들은 아직 출시할 시점이 아니기 때문
- 이런 경우 cherry-pick 기능을 사용

Cherry-pick 실습



Cherry-pick 실습

- 아래와 같은 상황에서 시작
- master에 있는 Commit4 커밋을 branch1의 최신 커밋으로 가져오는 것이 목표



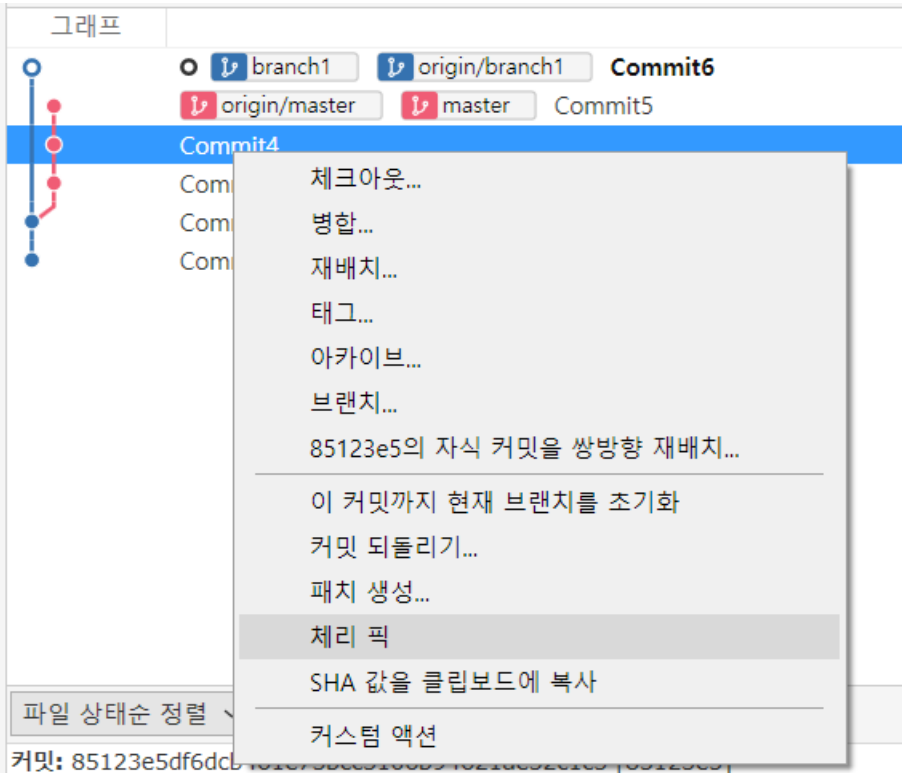
Cherry-pick 실습

- 커밋 내용

	GitFile1.txt	GitFile2.txt	GitFile3.txt
Commit6	Contents2	X	Contents1
Commit5	Contents4	Contents1	X
Commit4	Contents3	Contents1	X
Commit3	Contents3	X	X
Commit2	Contents2	X	X
Commit1	Contents1	X	X

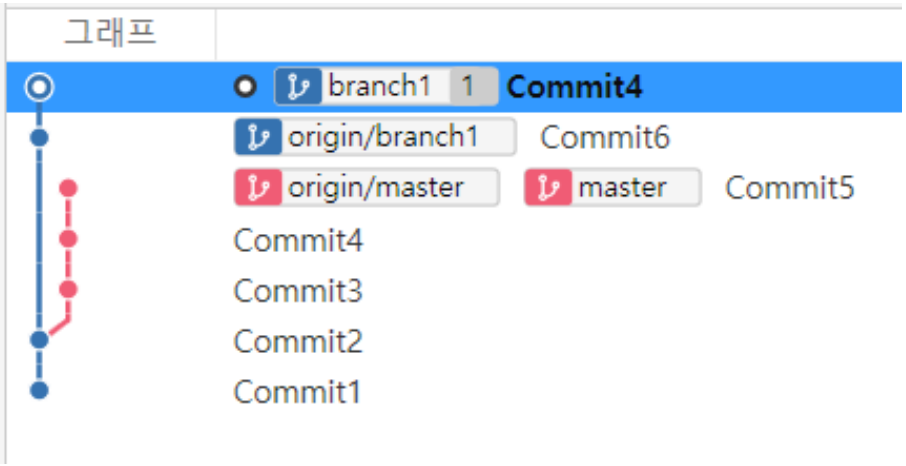
Cherry-pick 실습

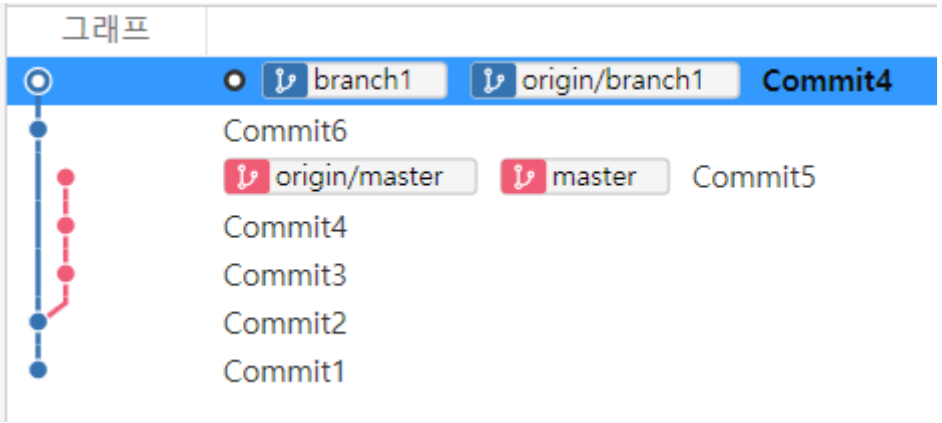
- 현재 브랜치에서 복제하길 원하는 커밋(다른 브랜치) 위에서 마우스 오른쪽 클릭을 한 후 [체리 픽]을 선택함
- 안내문이 뜨면 [확인] 버튼을 클릭함



Cherry-pick 실습

- 그러면 성공적으로 cherry-pick이 완료됨
- push까지 하면 원격저장소에 반영됨



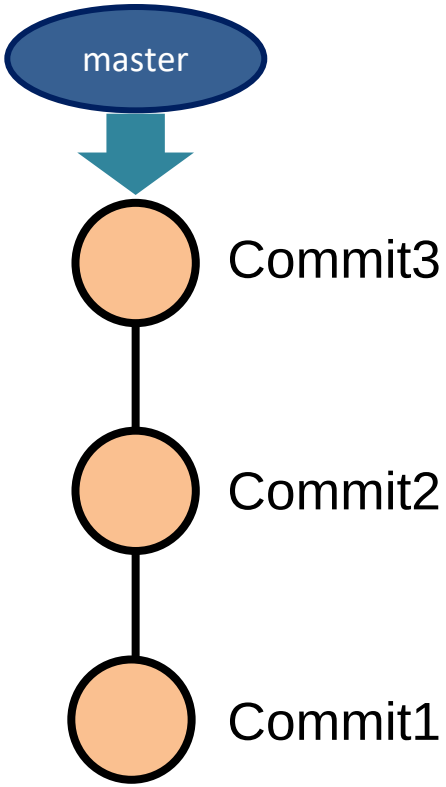


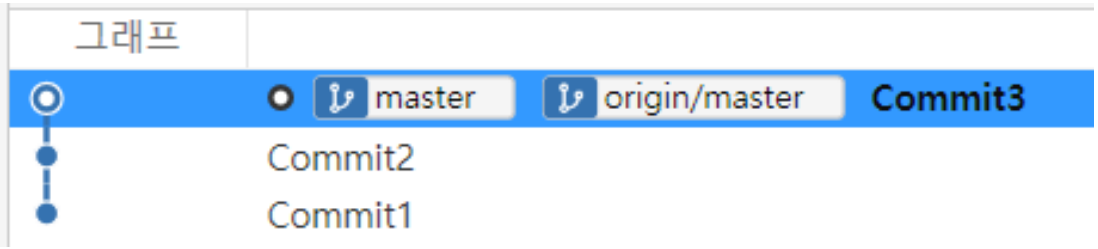
Reset

- 옛날 커밋으로 상태를 돌리고 싶을 때가 있음
- 이럴 때 쓰는 명령어가 reset
- 마치 없었던 일처럼 기억을 지우며 과거로 돌아가는 것

Reset 실습

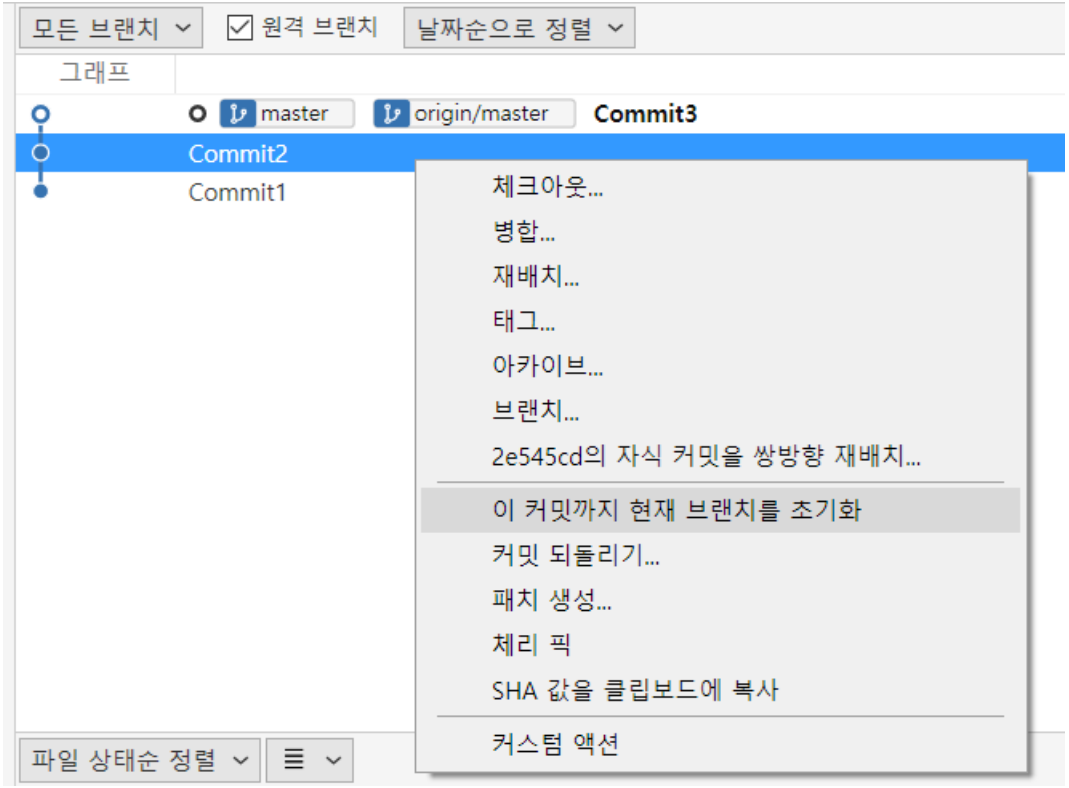
- 그림과 같이 master 브랜치에 커밋이 3개 되었고 push 까지 된 상황





Reset 실습

- master 브랜치의 최근 커밋을 Commit2 커밋으로 돌리고자 하는 상황
- 원하는 커밋(Commit2)에서 마우스 오른쪽 버튼 클릭한 후 [이 커밋까지 현재 브랜치를 초기화]를 클릭

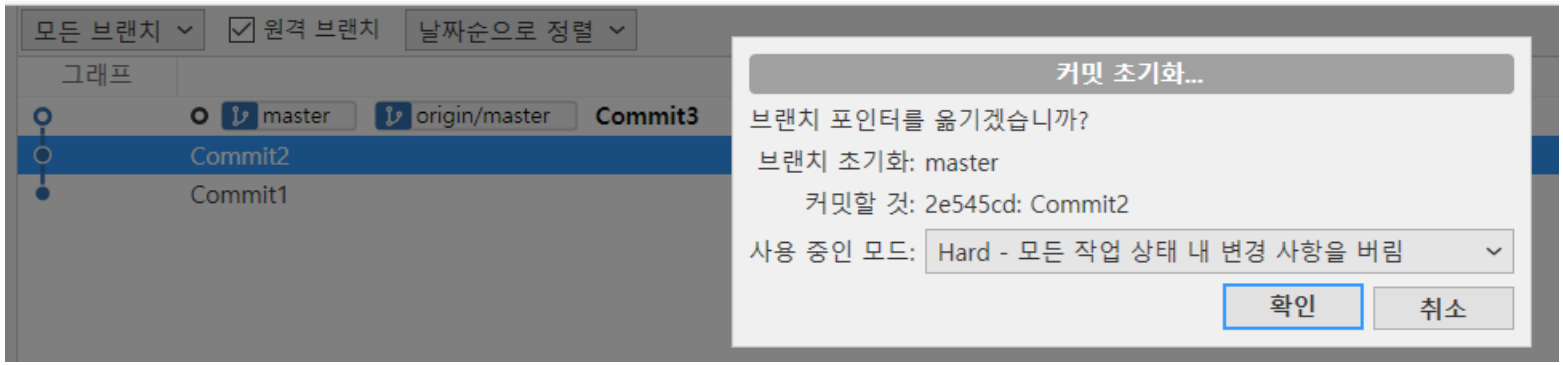


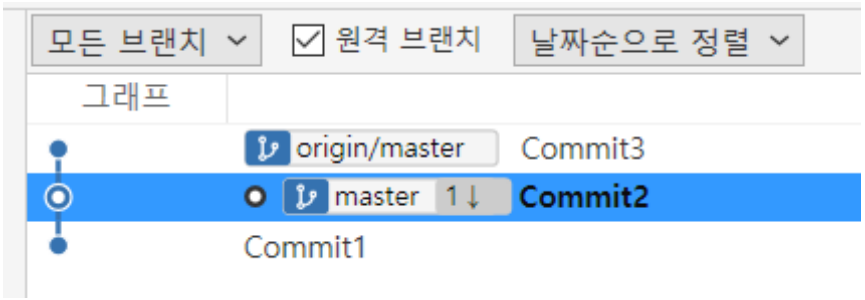
Reset 옵션

- Hard
  - 보통 하고 싶은 깔끔한 리셋
  - 지금 작업 공간이 더럽든, 깨끗하든, 혹은 이 커밋 다음에 100개의 커밋을 추가했든, 그냥 깔끔하게 히스토리를 돌리는 것임
- Mixed
  - 원하는 커밋으로 이력을 되돌리긴 하지만, 이 다음에 추가했던 모든 변경사항을 작업 공간으로 뽑아줌
- Soft
  - Mixed와 거의 동일
  - 다만, [Mixed] 모드는 변경사항을 스테이지 아래로 두어서 다시 무엇을 add할지 고민할 수 있음
  - Soft 모드는 변경사항을 스테이지 위로 두어서 당장 커밋 가능

Reset 실습

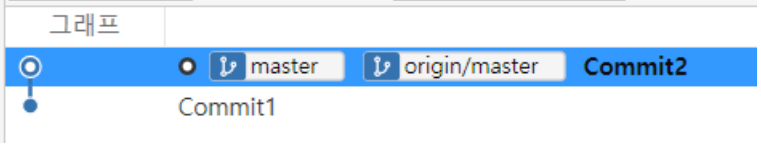
- Hard 모드로 reset한다고 함
- 경고창이 뜨면 [예] 클릭
- 그러면 정상적으로 reset이 이루어짐





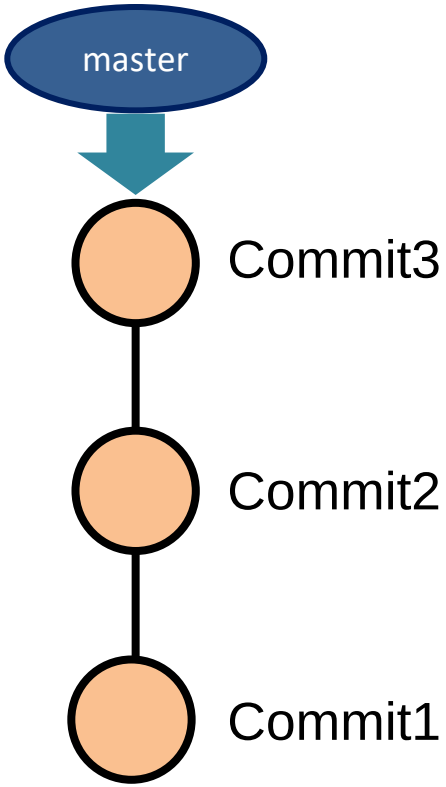
Reset 실습

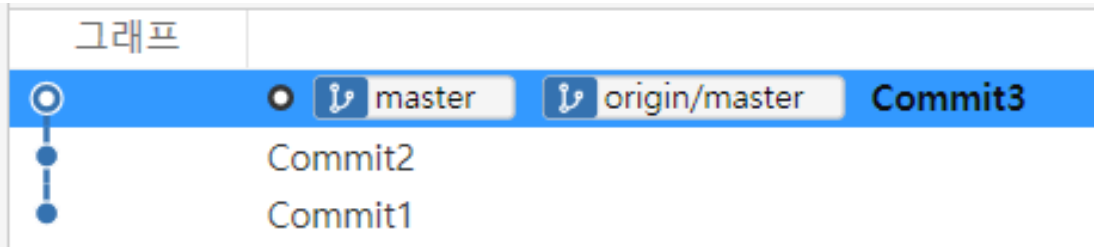
- 아직 원격저장소에는 커밋이 남아있음
- 변경사항을 원격저장소에도 반영하려면 강제 푸시를 해야함
- [강제 푸시] 체크박스 체크한 후 push
- 그러면 원격저장소도 성공적으로 reset이 됨



Reset 실습 - Mixed 모드

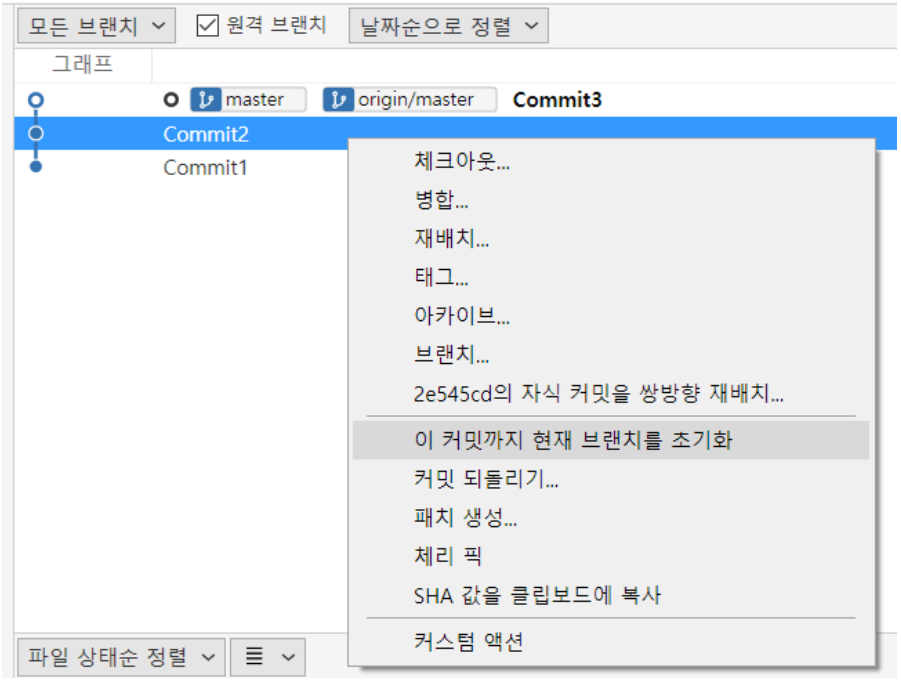
- 방금 Reset 실습하기 직전 상황으로 다시 돌아감
- 이번에는 mixed 모드로 reset을 하고자 함

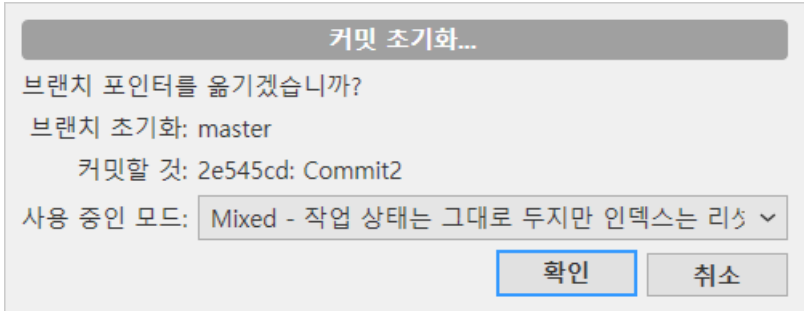




Reset 실습 - Mixed 모드

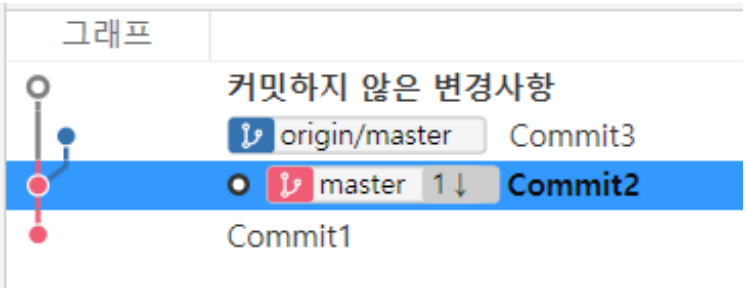
- 동일하게 Commit2 커밋으로 reset을 하되 mixed 모드를 선택함

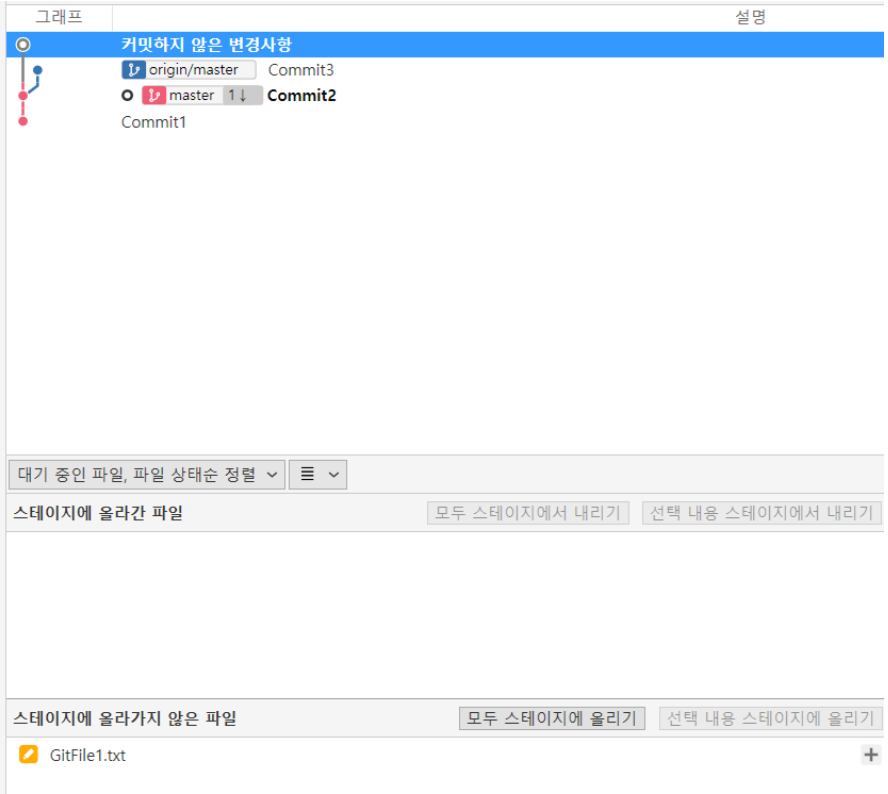




Reset 실습 - Mixed 모드

- 성공적으로 reset이 되었지만 현재 작업 공간에는 최근 내용이 그대로 남아있음
- 즉, GitFile1.txt 내용을 보면 Contents3임
- 또한 현재 GitFile1.txt 변경사항은 스테이지 아래에 있음



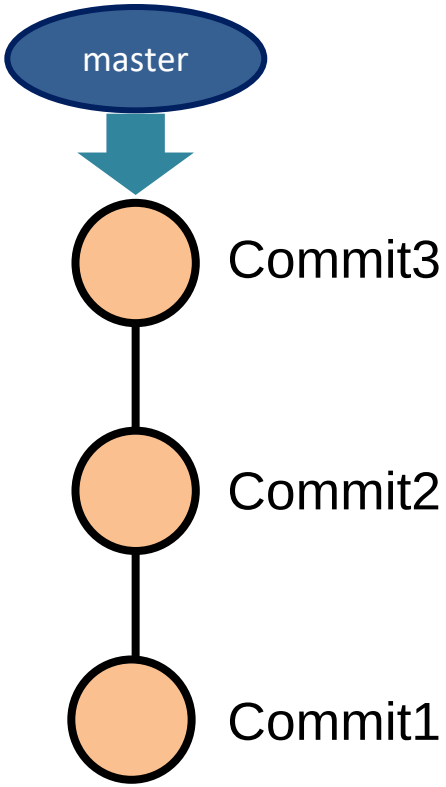


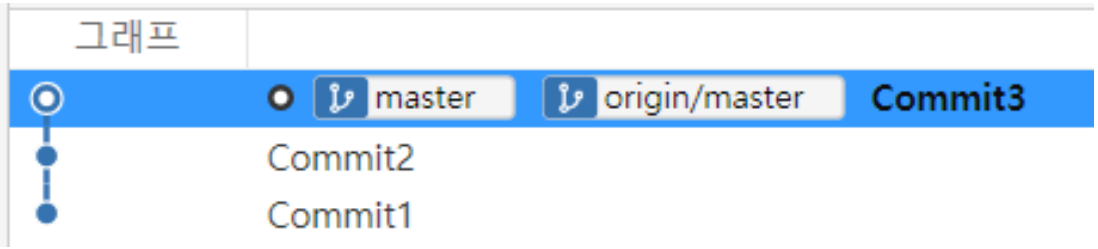
Revert

- reset으로 커밋을 마치 없었던 일처럼 되돌리는 것은 가능하지만 모두가 함께 쓰는 브랜치라 이력 관리가 중요하다면 변경사항을 되돌리는 새로운 커밋을 만드는 것이 더 좋음
- 이를 위한 기능이 revert임

Revert 실습

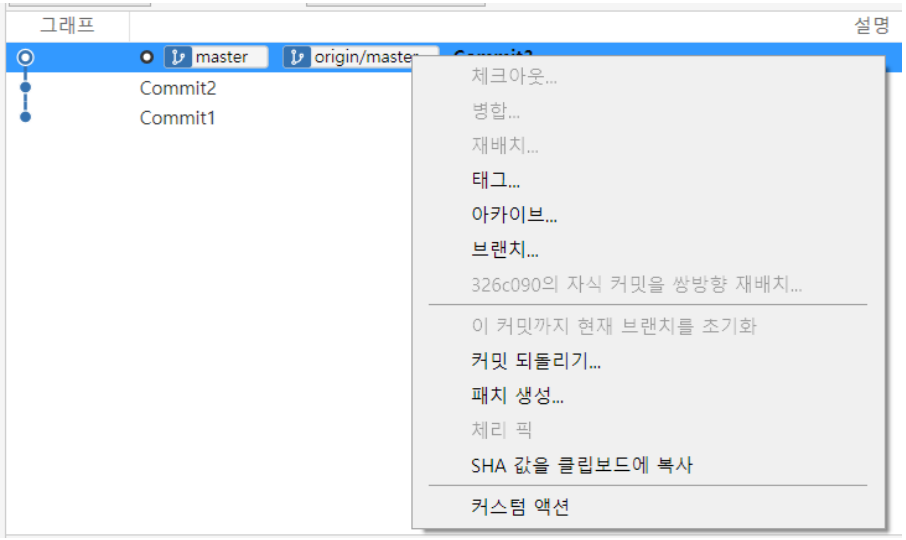
- 아래와 같이 커밋이 세 개 된 상태에서 시작함





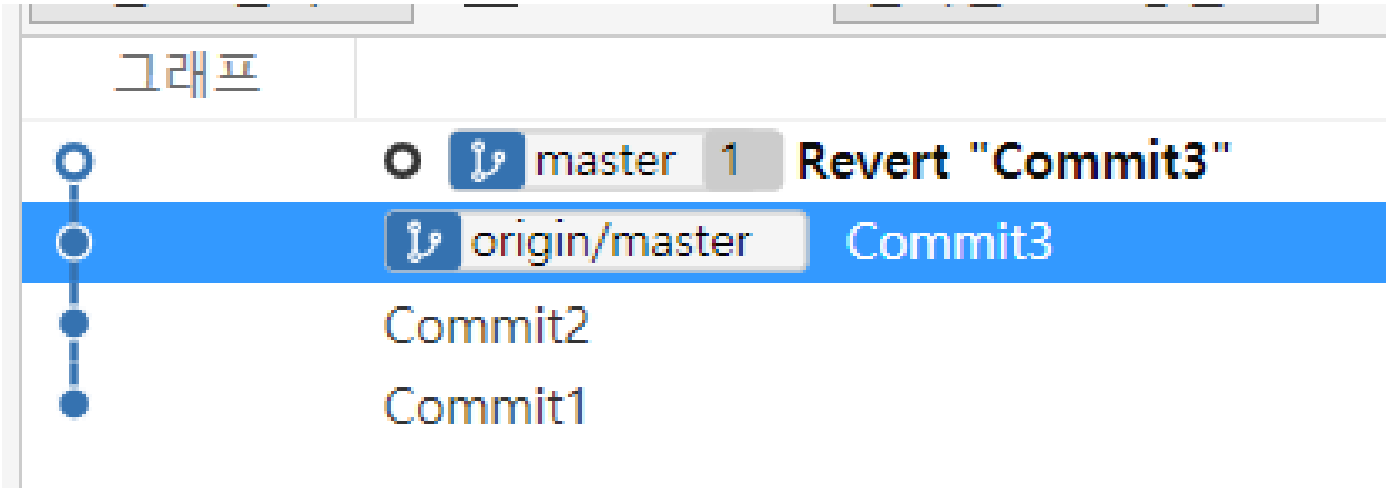
Revert 실습

- Commit3를 되돌리는 revert를 하고자 함
- 되돌리려는 커밋(Commit3)에서 마우스 오른쪽 클릭한 후 [커밋 되돌리기]를 선택함
- 확인 창에서 예를 클릭함



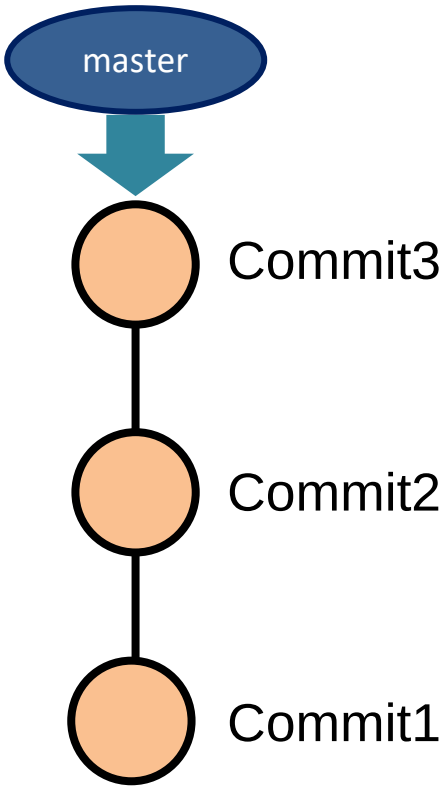
Revert 실습

- 성공적으로 Commit 커밋을 revert함
- Revert "Commit3"이라는 이름의 커밋이 새로 만들어짐
- 현재 GitFile1.txt의 내용은 Contents2인 것을 보아 Commit3 변경사항이 취소됨을 알 수 있음



Revert 실습

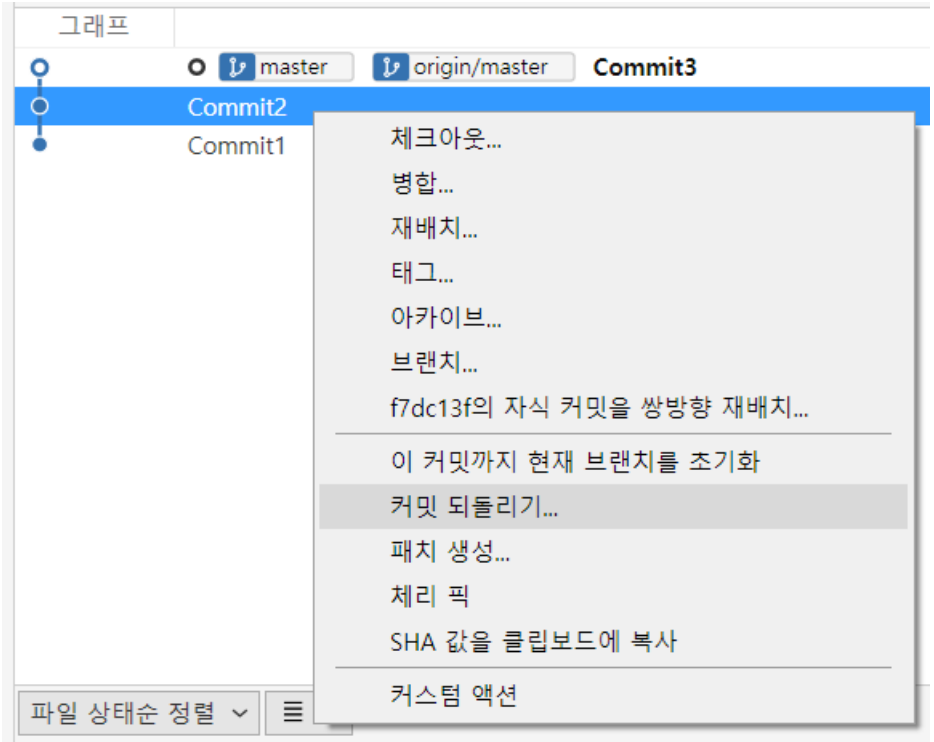
- 가장 최근 커밋이 아닌 옛날 커밋도 revert할 수 있음
- 아래와 같은 상황에서 Commit2 변경사항을 되돌릴 것임

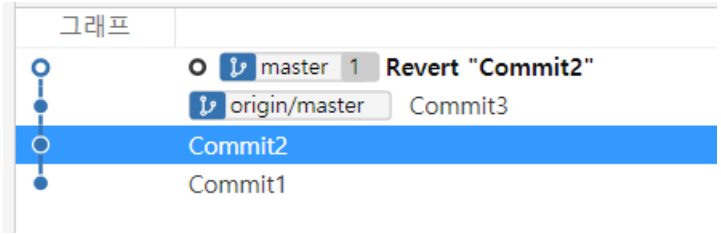


	GitFile1.txt	GitFile2.txt
Commit3	Contents2	Contents1
Commit2	Contents1	Contents1
Commit1	Contents1	X

Revert 실습

- 취소하려는 커밋(Commit2)에서 마우스 오른쪽 클릭하여 [커밋 되돌리기] 클릭
- 그러면 정상적으로 Revert "Commit2"가 이루어짐
- 현재 작업 공간을 보면 GitFile2.txt가 사라진 것을 확인





Revert 실습

	GitFile1.txt	GitFile2.txt
Revert "Commit2"	Contents2	X
Commit3	Contents2	Contents1
Commit2	Contents1	Contents1
Commit1	Contents1	X

Revert 충돌

- 어떤 커밋을 revert하고자 할 때, 그 커밋 이후에 동일한 부분을 추가 수정한 커밋이 있다면 충돌이 발생

파일내용	GitFile1.txt
Commit3	Contents3
Commit2	Contents2
Commit1	Contents1

- 가령 위 표와 같이 커밋한 상태에서 Commit2를 rever한다면 충돌이 발생함.
  - Commit2: Contents1->Contents2로 수정
  - Commit3: Contents2->Contents3로 수정
- 최신 커밋의 파일 내용이 Contents3인데 Commit2를 revert하려면 충돌 발생

Revert 충돌

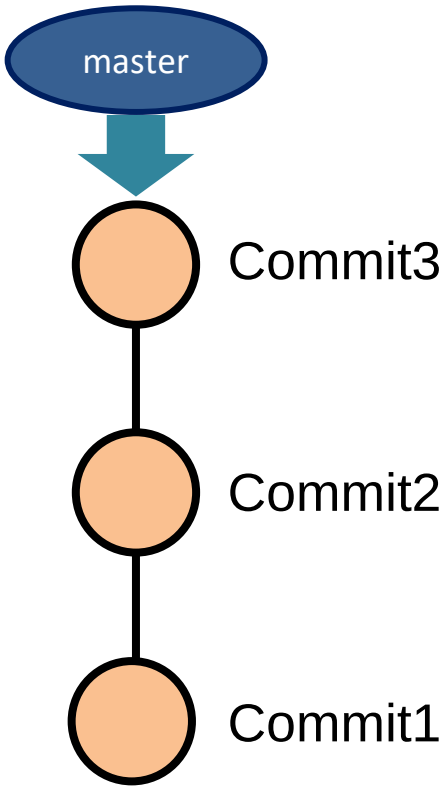
- 따라서, 동일한 부분에 대한 여러 커밋을 revert해야하는 경우 최신 커밋부터 revert를 하는 것이 좋음
- ex) master 브랜치가 C1, F1, C2, F2, C3 순으로 커밋이 되었다고 함. F1, F2 커밋을 revert하고자 함
- F2 revert -> F1 revert 순으로 진행
  - C1, F1, C2, F2, F3, RF2, RF1 (master)

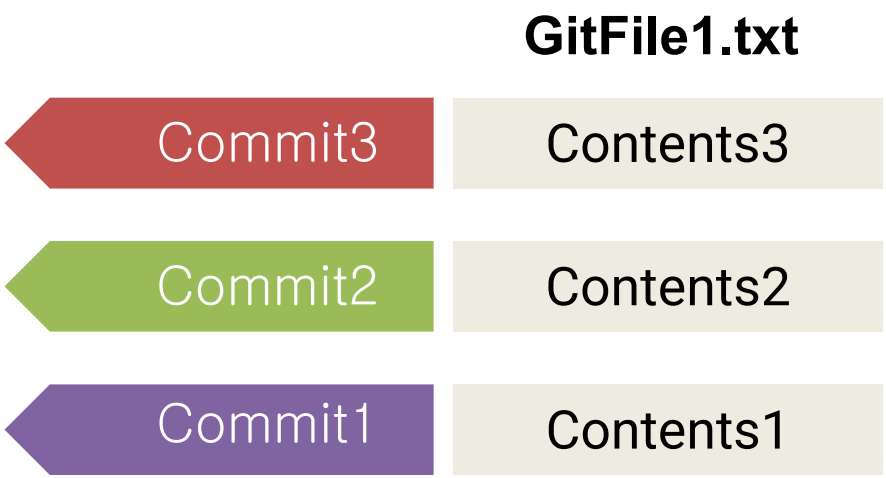
Stash

- 열심히 작업한 것이 있는데 급하게 다른 브랜치로 이동해야하는 상황
- 그런데 아직 커밋하지 않은 변경사항이 많이 있고 당장 커밋으로 만들기는 애매한 파일들임
- 이 변경사항을 잠깐 서랍속에 넣어두었다가 나중에 다시 꺼내 쓰는 방법이 stash
- stash에는 tracked 상태(추적중)인 파일들만 올라가고 새로 만든 파일은 untracked 상태이므로 들어가지 않음

Stash 실습

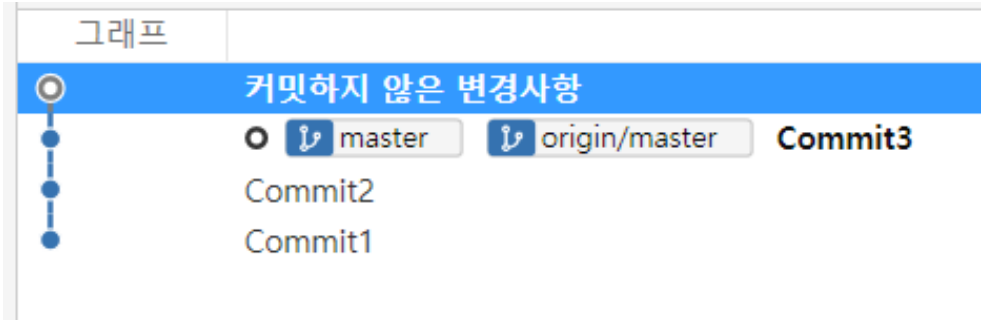
- 아래와 같은 상황에서 stash를 저장하는 실습 진행함



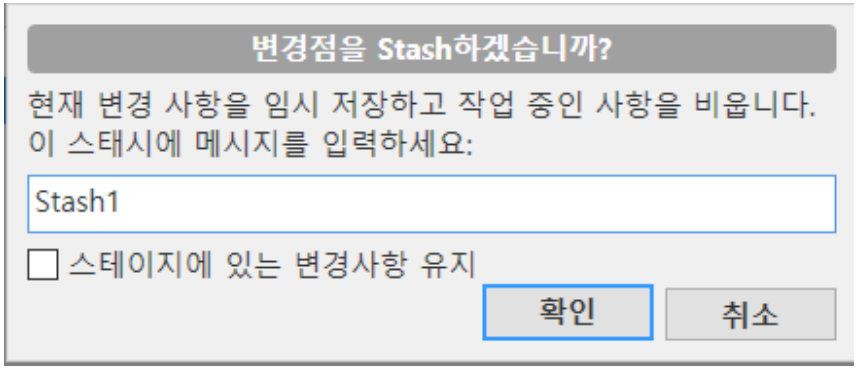


Stash 실습

- GitFile1.txt의 내용을 Contents4로 수정. 커밋하지는 않음

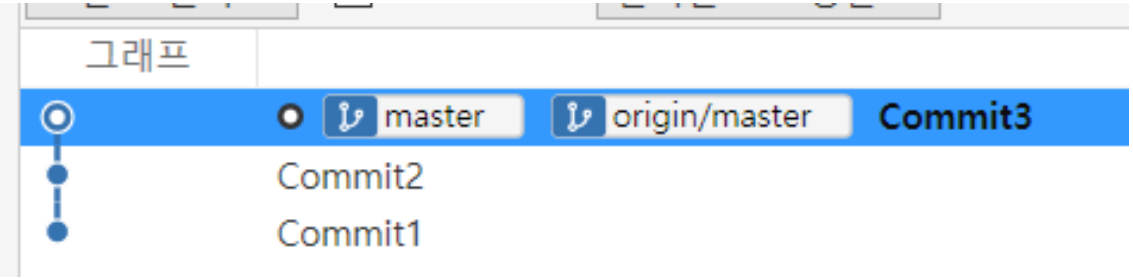


- 이제, 소스트리 창 상단에 [스태시] 아이콘을 클릭하고 스태시에 대한 설명을 적음
  - 예) Stash1



Stash 실습

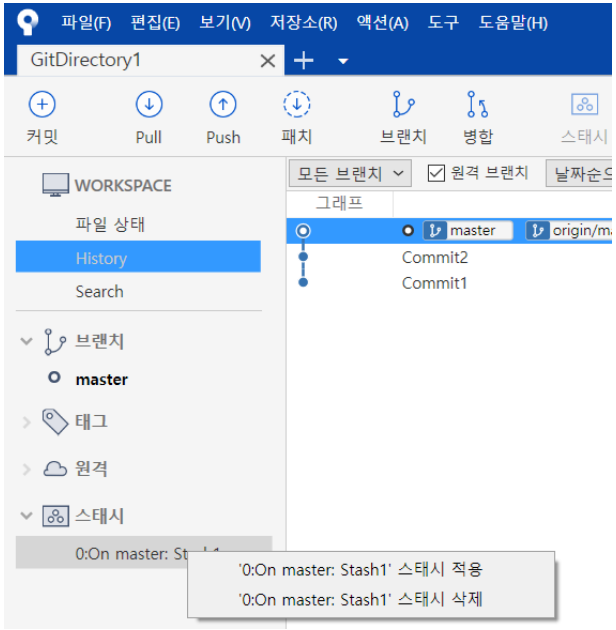
- 그 후 [확인] 버튼을 클릭하면 아래와 같이 작업공간이 깨끗해짐

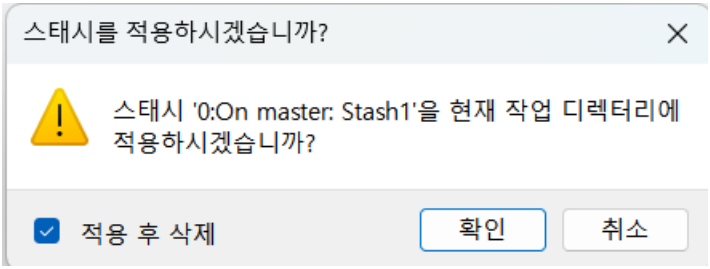


- GitFile1.txt 현재 내용을 보면 Contents3으로 되어있음. 즉, 최근 커밋3의 상태로 돌아감을 알 수 있음

Stash 실습

- 이번에는 저장한 stash를 꺼내는 것을 실습함
- 소스트리 왼쪽 사이드바의 [스태시] 섹션에서 원하는 스태시를 선택하고 마우스 오른쪽 클릭하여 [스태시 적용]을 클릭함
- [확인]을 누름. 만약 stash를 계속 쓸 것이 아니라면 [적용 후 삭제]를 체크하고 [확인]을 클릭하면 됨





Stash 실습

- 아까 저장해두었던 stash를 다시 가져옴
- 현재 GitFile1.txt의 내용을 보면 Contents4가 적혀있음을 확인할 수 있음. 즉, 이전에 작업해두었던 내용을 가져온 것임