

# git 공부

HEAD드러내기 : # git checkout main

# git checkout [Hash value]

→ 해쉬값은 앞 4자리값만 쳐줘도 된다.

상대참조(^) : 한번에 한 커밋 위로 움직이는것

# git checkout main^

# git checkout [Hash Value]^

# git checkout HEAD^;

상대참조(~) : 한번에 여러 커밋 위로 움직이는것

# git branch -f main HEAD~3 → main head에서 3칸 위로 올라감

# git branch -f [branch name] [Hash value]

# git checkout HEAD~4

git 작업 되돌리기 → reset, revert

reset : 애초에 커밋하지 않은것처럼 예전 커밋으로 브랜치를 옮기는것이다. (단, 협업시에는 비추천)

→ 다른사람과 공유가 되지않기때문

revert : 변경분을 되돌리고 사람들과 공유가 가능

# git reset HEAD~1 → 1개 커밋을 되돌린다.(삭제)

# git revert HEAD → 현재의 커밋을 되돌린다.(복원)

차이점

### Git 리셋(reset)

`git reset`은 브랜치로 하여금 이전의 커밋을 가리키도록 이동시키는 방식으로 변경 내용을 되돌립니다. 이런 관점에서 "히스토리를 고쳐쓴다"라고 말할 수 있습니다. 즉, `git reset`은 마치 예전에 커밋하지 않은 것처럼 예전 커밋으로 브랜치를 옮기는 것입니다.

어떤 그림인지 한번 보죠:

```
git reset HEAD~1
```

### Git 리버트(revert)

각자의 컴퓨터에서 작업하는 로컬 브랜치의 경우 리셋(reset)을 할 수 있습니다만, "히스토리를 고쳐쓴다"는 점 때문에 다른 사람이 작업하는 리모트 브랜치에는 할 수 없습니다.

변경분을 되돌리고, 이 되돌린 내용을 다른 사람들과 공유하기 위해서는, `git revert`를 써야합니다. 예제로 살펴볼게요.

```
git revert HEAD
```