

Commit

- 저장 언제든 commit한 시점으로 되돌아 갈 수 있음
- commit 하고 싶은 것들만 (파일을 묶음)
add or stage에 올림.
- commit 메시지 짓을 간략, 한글 띄우고 자세히

- But, 내 컴퓨터에 저장 됨.
다라서 git에 저장 = push (업로드)

Source Tree

- git은 원래 명령어 사용 But, 어렵!
- GUI (편하게 사용위해서)
- Clone > 내 컴퓨터로 가져옴.

Clone - add - commit - push

checkout

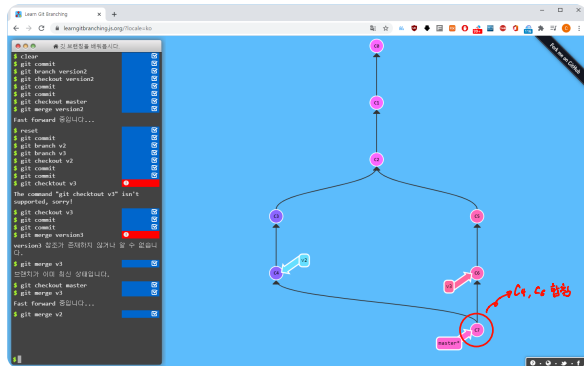
- 되돌아 가기. (특정 branch로 이동함)
- Source Tree : 코드 관리 체계.

branch

- Snapshot 과 비슷 함됨.
ex) 새로운 Snapshot을 만드는 것과 유사.
- 현재 branch = Head branch

Merge

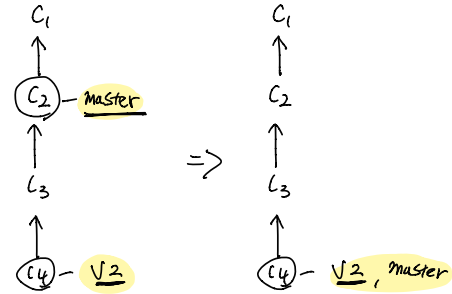
- branch를 합침.



✓ 충돌 가능. - ?

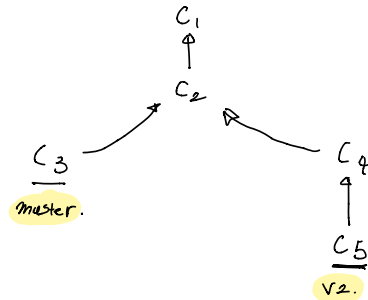
- merge 후 필요없다면 삭제

1.



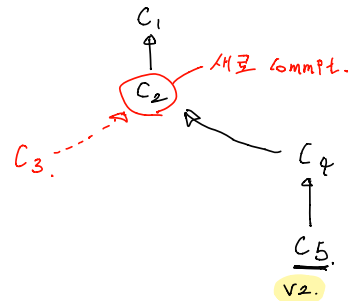
=> fast-forward : 단순히 갱신 됨.

2.

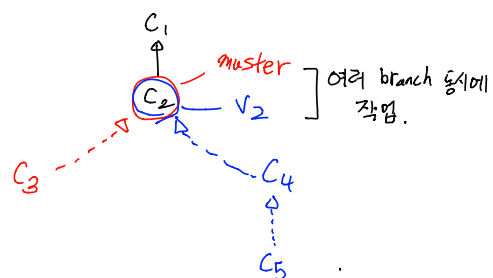


① C2 (과거)에서 branch 생성 & commit.

②



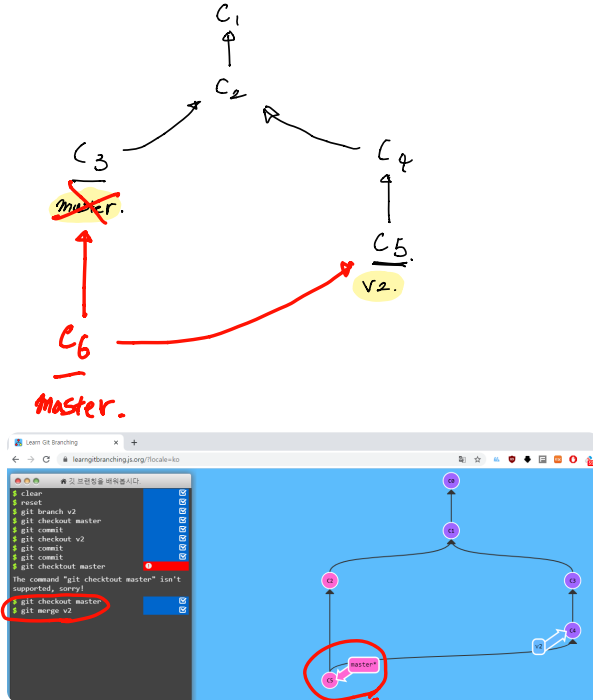
③



=> 여러 branch에서 동시 변경 = 충돌.

∴ 최신 내용 (주로 target branch에 있음)

충돌 발생 하지 않았을 때.



Source Tree - pull

- 현재보다 최신 Commit이 서버에 있음. (다들 사용해서 자크체크)
pull시 합쳐줌.

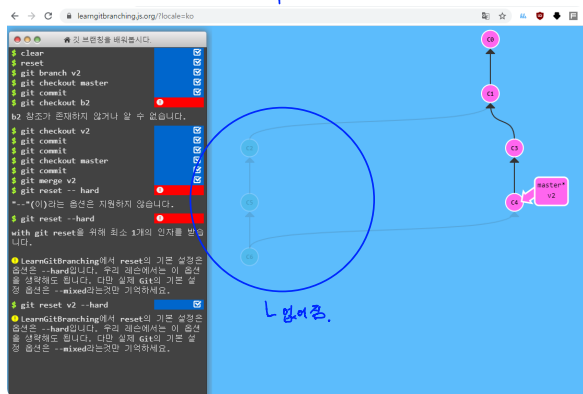
* 충돌시

1. 숙동
2. 저장소 or 니 것.

reset

- Commit 날라갈 수 있다.

git reset v2 --hard.

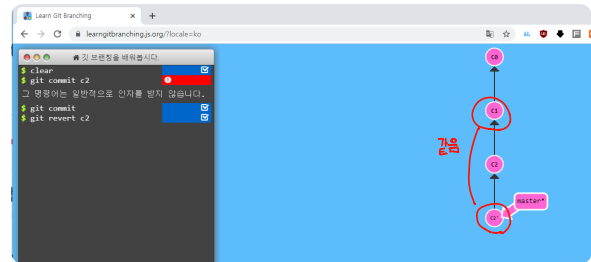


branch로 Commit 되들름.

- 쉽게만 트리 복원. 내용 살아있음.
- ⇒ git checkout branch.

revert.

- Commit 보존 하면서 내용만 되들름.



- Commit이 하지 않은 변경사항 있을 경우 Checkout이 안됨

- ① branch 1에서 임시 Commit.
- ② branch 2로 checkout. → 하변작업 계속...
- ③ branch 1로 돌아와서 1 작업 마무리.

Source Tree - 마지막 커밋 과정.

stash 사용.

- 임시저장공간.
- (?) 신규파일 → stash 만들 why? 변경사항 X
- 이전 Commit 되게시키기.

Rebase (재배치)

- branch 합침. (현재 branch가 대상 branch 위로)
- Commit history 깔끔 But, 위험.

git 기초 명령어

- clone → 원격 저장소 복사.
- add → stage 영역에 파일 추가.
- Commit → 저장
- Push → 업로드.