## 13주차 In-clsss 실습 과제 시트

강 환수 교수

|               | reset & revert: 재설정과 취소 [교재 9장 복귀]  |                                |                                       |  |  |  |  |
|---------------|---|--------------------------------|---------------------------------------|--|--|--|--|
|               | □ reset과 revert 이해와 활용  |                                |                                       |  |  |  |  |
| 프로젝트          | O git reset HEAD~   |                                |                                       | # Git Graph × □ · · · Show Remote Branches Q ► ★ Ở |  |  |  |
|               | O git revert HEAD   |                                | Branches: Show All  Graph Description | ▼ Show Remote Branches Q                           |  |  |  |
|               |   | ○ revert 충돌 발생시, 파일 수정 후 추가/커밋 |                                       |  |  |  |  |
|               | O reset 옵션softmixed,hard  |                                |                                       |  |  |  |  |
| 모듈1           | □ 저장소 uplog 생성, 브랜치 main에서 다음의 3개의 커밋 생성  |                                |                                       |  |  |  |  |
|               | □ 브랜치 main에서 revert로 커밋 B 상태로 이동 후 다시 원 상태로 복귀                                    |                                |                                       |  |  |  |  |
|               | O 커밋 C를 취소해야 커밋 B로 이동   |                                |                                       |  |  |  |  |
| 모듈2           | • \$ git revert HEADno-edit   |                                |                                       |  |  |  |  |
|               | ○ 다시 커밋 A, B, C 상태로 복귀  |                                |                                       |  |  |  |  |
|               | • \$ git resethard <커밋C_id>   |                                |                                       |  |  |  |  |
|               | □ 브랜치 main에서 re   | set으로 커밋 B 상태로                 | 이동 후 다시 원 상태                          | 태로 복귀  |  |  |  |
| 모듈3           | O reset으로 커밋 B(HEAD~)로 이동   |                                |                                       |  |  |  |  |
|               | •hard로 이동하면 모든 3-공간이 커밋 내용으로 수정   |                                |                                       |  |  |  |  |
|               | • \$ git resethard HEAD~  |                                |                                       |  |  |  |  |
|               | ○ 다시 커밋 A, B,   | C 상태로 복귀                       |                                       |  |  |  |  |
|               | • \$ git reset  | hard ORIG_HEAD                 |                                       |  |  |  |  |
|               | □ 브랜치 main에서 re   | vert로 커밋 C 상태로                 | 이동 후 다시 원 상태                          | 로 복귀   |  |  |  |
|               | ○ 커밋 B를 취소해   | 야 커밋 C로 이동                     |                                       |  |  |  |  |
|               | • \$ git revert HEAD~   |                                |                                       |  |  |  |  |
| <br>모듈4       | ○ 충돌 발생 시, h.txt을 열어 파일을 수정하고 다시 추가 커밋 수행   |                                |                                       |  |  |  |  |
| 포괄4<br>       | \$ vscode h.txt   |                                |                                       |  |  |  |  |
|               | • \$ git commit –am 'Revert B, move C'  |                                |                                       |  |  |  |  |
|               | ○ 다시 커밋 A, B, C 상태로 복귀  |                                |                                       |  |  |  |  |
|               | • \$ git resethard <커밋C_id>   |                                |                                       |  |  |  |  |
| 결과 PR로        | ✓ QB-01-20221234-홍길동-13주차-결과.md   |                                |                                       |  |  |  |  |
| 보내기           | · VD VI 20221204 중단 8 IST 시 선택.IIIU   |                                |                                       |  |  |  |  |
|               | □ 기준 폴더 13w에서 시작  |                                |                                       |  |  |  |  |
| 준비            | \$ git configglobal core.editor 'codewait'  |                                |                                       |  |  |  |  |
|               | \$ git configglobal core.autocrlf true<br>\$ git configglobal core.safecrlf false |                                |                                       |  |  |  |  |
|               | □ 저장소 uplog 생성, 브랜치 main에서 다음의 3개의 커밋 생성  |                                |                                       |  |  |  |  |
|               | 브랜치   | main                           | main                                  | main   |  |  |  |
|               |   | HEAD^^                         | HEAD^                                 |  |  |  |  |
|               | HEAD  | HEAD~2                         | HEAD∼1                                | HEAD   |  |  |  |
| 모듈1           | 커밋 메시지 파일   | A<br>h tvrt                    | B<br>b twt                            | C  |  |  |  |
| _ 포플!<br>  실습 | - 박일<br>  | h.txt                          | h.txt                                 | h.txt<br>aaa                                       |  |  |  |
| 20            | 내용  | aaa                            | aaa<br>bbb                            | bbb<br>ccc   |  |  |  |
|               |   |                                |                                       |  |  |  |  |
|               | PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w<br>\$ git init uplog                    |                                |                                       |  |  |  |  |
|               | Initialized empty Git repository in C:/OSS/git/13w/uplog/.git/                    |                                |                                       |  |  |  |  |
|               | PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w<br>\$ cd uplog                          |                                |                                       |  |  |  |  |

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main) \$ echo aaa > h.txt

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main) \$ git lg1 3ce2309 (HEAD -> main) C 8370ac8 B d636259 A

- □ 브랜치 main에서 revert로 커밋 B 상태로 이동 후 다시 원 상태로 복귀
  - 커밋 C를 취소해야 커밋 B로 이동
    - \$ git revert HEAD --no-edit

| HEAD   | HEAD^^<br>HEAD~2 | HEAD^<br>HEAD~1 | HEAD              |
|--------|------------------|-----------------|-------------------|
| 커밋 메시지 | A                | В С             |                   |
| 파일     | h.txt            | h.txt           | h.txt             |
| 내용     | aaa              | aaa<br>bbb      | aaa<br>bbb<br>ccc |

- 다시 커밋 A, B, C 상태로 복귀
  - \$ git reset --hard <커밋C\_id>

| HEAD   | HEAD^^^<br>HEAD~3 | HEAD^^<br>HEAD~2 | HEAD^<br>HEAD~1   | HEAD       |
|--------|-------------------|------------------|-------------------|------------|
| 커밋 메시지 | A                 | В                | С                 | Revert "C" |
| 파일     | h.txt             | h.txt            | h.txt             | h.txt      |
| 내용     | aaa               | aaa<br>bbb       | aaa<br>bbb<br>ccc | aaa<br>bbb |

## 모듈2 실습

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)

Sgit Ig1 3ce2309 (HEAD -> main) C 8370ac8 B d636259 A

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)

\$ git revert HEAD --no-edit [main 51dcbe0] Revert "C"
Date: Sun Nov 13 08:13:48 2022 +0900 1 file changed, 1 deletion(-)

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)

\$ git lg1 51dcbe0 (HEAD -> main) Revert "C" 3ce2309 C 8370ac8 B d636259 A

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)

\$ cat h.txt

aaa bbb

ccc

```
□ 브랜치 main에서 reset으로 커밋 B 상태로 이동 후 다시 원 상태로 복귀
              O reset으로 커밋 B(HEAD~)로 이동
                  • --hard로 이동하면 모든 3-공간이 커밋 내용으로 수정
                  • $ git reset --hard HEAD~
              ○ 다시 커밋 A, B, C 상태로 복귀

    $ git reset --hard ORIG_HEAD

           PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
           $ git lg1
3ce2309 (HEAD -> main) C
8370ac8 B
           d636259 A
           PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
모듈3
           $ git reset --hard HEAD~
HEAD is now at 8370ac8 B
실습
           PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
           $ git lg1
           $ git 1g1
8370ac8 (HEAD -> main) B
d636259 A
           PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
           $ cat h.txt
           bbb
           □ 브랜치 main에서 revert로 커밋 A 상태로 이동 후 다시 원 상태로 복귀
              ○ 커밋 B를 취소해야 커밋 A로 이동
                  • $ git revert HEAD~
              O 충돌 발생 시, h.txt을 열어 파일을 수정하고 다시 추가 커밋 수행

    $ code h.txt

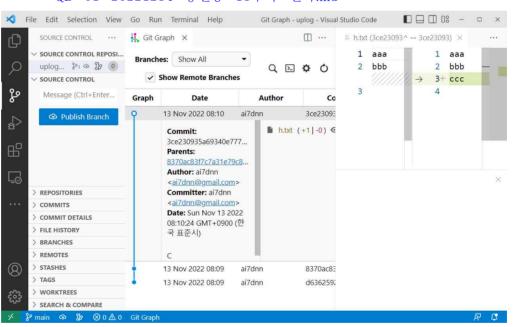
                  • $ git commit –am 'Revert B, move A' 또는 add 후 revert --continue
              ○ 다시 커밋 A, B, C 상태로 복귀
                  • $ git reset --hard <커밋C_id>
           PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
           3ce2309 (HEAD -> main) C
8370ac8 B
           $ git lg1
모듈4
실습
           d636259 A
           PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
           $ git revert HEAD~
           Auto-merging h.txt
           CONFLICT (content): Merge conflict in h.txt error: could not revert 8370ac8... B
          hint: After resolving the conflicts, mark them with hint: "git add/rm <pathspec>", then run hint: "git revert ——continue".
hint: You can instead skip this commit with "git revert ——skip".
           hint: To abort and get back to the state before "git revert",
           hint: run "git revert --abort"
           PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main|REVERTING)
           $ code h.txt
```

C: > OSS > git > 13w > uplog > ≡ h.txt 1 aaa Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes <<<<< HEAD (Current Change) bbb 4 ccc 5 \_\_\_\_ >>>>> parent of 8370ac8 (B) (Incoming Change) [Accept Incoming Change]를 눌러 내용을 aaa로 수정, 저장 후 닫기 PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main) \$ git lg1 3ce2309 (HEAD -> main) C 8370ac8 B # 다음 방법으로 PR(pull request) 해 주세요 □ 팀장 • 팀 계정에서 지금 우리 수업의 Forked된 저장소를 새로 Sync한 후

## □ 팀원

- 자신의 계정에서 Forked된 팀 저장소를 새로 Sync한 후
- Forked된 팀 저장소의 이 폴더에서 다음 파일을 생성해 vscode 화면을 추가한 후
- vscode에서 Git graph와 커밋 C에서 파일의 차이를 열어 모두 보이게 화면 구성 ✓ QB-01-20221234-홍길동-13주차-결과.md

결과 PR로 보내기



• Forked된 팀 저장소에서 PR로 송부하고

## □ 팀장

• 다시 팀장은 우리 수업의 Forked된 저장소에서 팀원의 자료를 병합(merge)한 후 우리 수업의 Forked된 저장소에서 PR로 송부

