

13주차 In-class 실습 과제 시트(정답)

강 환수 교수

프로젝트	<div>reset & revert 재설정과 취소 [교재 9장 복귀]</div> <div><div><div><div><input type="checkbox"/> reset과 revert 이해와 활용</div><div><div><div><div>○ git reset HEAD~</div><div>○ git revert HEAD</div><div>○ revert 충돌 발생시, 파일 수정 후 추가/커밋</div><div>○ reset 옵션 --soft --mixed, --hard</div></div></div></div></div></div><div><div><div>Git Graph</div><div><div><div><div><div></div><div></div><div></div><div></div><div></div></div><div><div>main</div><div>B</div><div>A</div></div><div><div>C</div></div></div></div><div><div>13 Nov 2022 08:10</div><div>ai7dnn</div><div>3ce23093</div></div></div><div><div>13 Nov 2022 08:09</div><div>ai7dnn</div><div>8370ac83</div></div><div><div>13 Nov 2022 08:09</div><div>ai7dnn</div><div>d6362592</div></div></div></div></div>																				
	<div>모듈1</div> <div><div><input type="checkbox"/> 저장소 uplog 생성, 브랜치 main에서 다음의 3개의 커밋 생성</div></div>																				
모듈2	<div><div><input type="checkbox"/> 브랜치 main에서 revert로 커밋 B 상태로 이동 후 다시 원 상태로 복귀</div><div><div><div>○ 커밋 C를 취소해야 커밋 B로 이동</div><div><div><div>• \$ git revert HEAD --no-edit</div></div></div><div>○ 다시 커밋 A, B, C 상태로 복귀</div><div><div><div>• \$ git reset --hard <커밋C_id></div></div></div></div></div></div>																				
모듈3	<div><div><input type="checkbox"/> 브랜치 main에서 reset으로 커밋 B 상태로 이동 후 다시 원 상태로 복귀</div><div><div><div>○ reset으로 커밋 B(HEAD~)로 이동</div><div><div><div>• --hard로 이동하면 모든 3-공간이 커밋 내용으로 수정</div><div><div><div>• \$ git reset --hard HEAD~</div></div></div><div>○ 다시 커밋 A, B, C 상태로 복귀</div><div><div><div>• \$ git reset --hard ORIG_HEAD</div></div></div></div></div></div></div></div>																				
모듈4	<div><div><input type="checkbox"/> 브랜치 main에서 revert로 커밋 C 상태로 이동 후 다시 원 상태로 복귀</div><div><div><div>○ 커밋 B를 취소해야 커밋 C로 이동</div><div><div><div>• \$ git revert HEAD~</div></div></div><div>○ 충돌 발생 시, h.txt을 열어 파일을 수정하고 다시 추가 커밋 수행</div><div><div><div>• \$ vscode h.txt</div><div>• \$ git commit -am 'Revert B, move C'</div></div></div><div>○ 다시 커밋 A, B, C 상태로 복귀</div><div><div><div>• \$ git reset --hard <커밋C_id></div></div></div></div></div></div>																				
결과 PR로 보내기	<div><div>✓ QB-01-20221234-홍길동-13주차-결과.md</div></div>																				
준비	<div><div><input type="checkbox"/> 기준 폴더 13w에서 시작</div><div><div><div><div>\$ git config --global core.editor 'code --wait'</div><div>\$ git config --global core.autocrlf true</div><div>\$ git config --global core.safecrlf false</div></div></div></div></div>																				
모듈1 실습	<div><div><input type="checkbox"/> 저장소 uplog 생성, 브랜치 main에서 다음의 3개의 커밋 생성</div><div><table><tr><td>브랜치</td><td>main</td><td>main</td><td>main</td></tr><tr><td>HEAD</td><td>HEAD ^ ^ HEAD~2</td><td>HEAD ^ HEAD~1</td><td>HEAD</td></tr><tr><td>커밋 메시지</td><td>A</td><td>B</td><td>C</td></tr><tr><td>파일</td><td>h.txt</td><td>h.txt</td><td>h.txt</td></tr><tr><td>내용</td><td>aaa</td><td>aaa bbb</td><td>aaa bbb ccc</td></tr></table></div><div><div>PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w</div><div>\$ git init uplog</div><div>Initialized empty Git repository in C:/OSS/git/13w/uplog/.git/</div><div>PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w</div><div>\$ cd uplog</div></div></div>	브랜치	main	main	main	HEAD	HEAD ^ ^ HEAD~2	HEAD ^ HEAD~1	HEAD	커밋 메시지	A	B	C	파일	h.txt	h.txt	h.txt	내용	aaa	aaa bbb	aaa bbb ccc
브랜치	main	main	main																		
HEAD	HEAD ^ ^ HEAD~2	HEAD ^ HEAD~1	HEAD																		
커밋 메시지	A	B	C																		
파일	h.txt	h.txt	h.txt																		
내용	aaa	aaa bbb	aaa bbb ccc																		

```
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 add .

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
$ git commit -m A
[main (root-commit) d636259] A
1 file changed, 1 insertion(+)
create mode 100644 h.txt

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

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
$ git commit -am B
[main 8370ac8] B
1 file changed, 1 insertion(+)

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

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
$ git commit -am C
[main 3ce2309] C
1 file changed, 1 insertion(+)

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
$ git lg1 # git log --oneline
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 ^ ^ HEAD~2	HEAD ^ HEAD~1	HEAD
커밋 메시지	A	B	C
파일	h.txt	h.txt	h.txt
내용	aaa	aaa bbb	aaa bbb ccc

○ 다시 커밋 A, B, C 상태로 복귀

- \$ git reset --hard <커밋C_id>

HEAD	HEAD ^ ^ ^ HEAD~3	HEAD ^ ^ HEAD~2	HEAD ^ HEAD~1	HEAD
커밋 메시지	A	B	C	Revert "C"
파일	h.txt	h.txt	h.txt	h.txt
내용	aaa	aaa bbb	aaa bbb ccc	aaa bbb

```
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)
$ 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

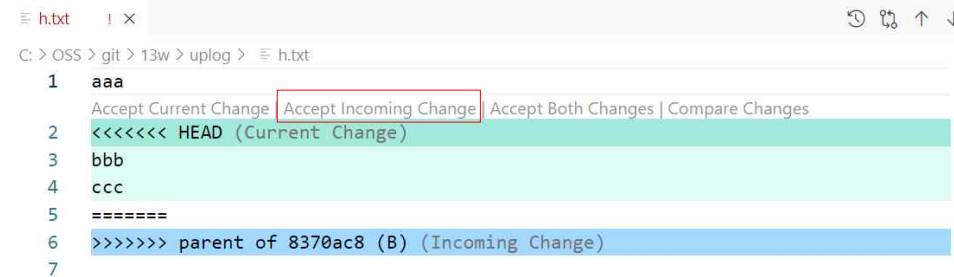
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
$ cat h.txt
aaa
bbb
```

모듈2
실습

	<pre>PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main) \$ git reset --hard HEAD^ HEAD is now at 3ce2309 C 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) \$ cat h.txt aaa bbb ccc</pre>
모듈3 실습	<div> <input type="checkbox"/> 브랜치 main에서 reset으로 커밋 B 상태로 이동 후 다시 원 상태로 복귀 <ul style="list-style-type: none"> ○ reset으로 커밋 B(HEAD~)로 이동 <ul style="list-style-type: none"> • --hard로 이동하면 모든 3-공간이 커밋 내용으로 수정 • \$ git reset --hard HEAD~ ○ 다시 커밋 A, B, C 상태로 복귀 <ul style="list-style-type: none"> • \$ git reset --hard ORIG_HEAD </div> <pre>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) \$ git reset --hard HEAD~ HEAD is now at 8370ac8 B PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main) \$ git lg1 8370ac8 (HEAD -> main) B d636259 A PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main) \$ cat h.txt aaa bbb PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main) \$ git reset --hard ORIG_HEAD HEAD is now at 3ce2309 C 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) \$ cat h.txt aaa bbb ccc</pre>
모듈4 실습	<div> <input type="checkbox"/> 브랜치 main에서 revert로 커밋 A 상태로 이동 후 다시 원 상태로 복귀 <ul style="list-style-type: none"> ○ 커밋 B를 취소해야 커밋 A로 이동 <ul style="list-style-type: none"> • \$ git revert HEAD~ ○ 충돌 발생 시, h.txt을 열어 파일을 수정하고 다시 추가 커밋 수행 <ul style="list-style-type: none"> • \$ code h.txt • \$ git commit -am 'Revert "B", move "A"' 또는 add 후 revert --continue ○ 다시 커밋 A, B, C 상태로 복귀 <ul style="list-style-type: none"> • \$ git reset --hard <커밋C_id> </div> <pre>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) \$ 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</pre>

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".

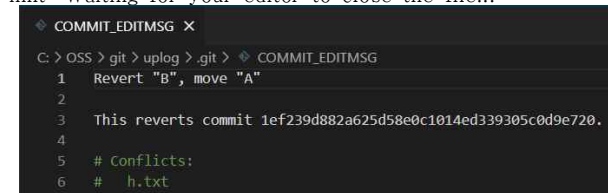
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main|REVERTING)
 \$ code h.txt



[Accept Incoming Change]를 눌러 내용을 aaa로 수정, 저장 후 닫기

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main|REVERTING)
 \$ git add h.txt

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main|REVERTING)
 \$ git revert --continue
 hint: Waiting for your editor to close the file...



add 후 revert --continue는 다음으로도 가능
 \$ git commit -am 'Revert "B", move "A"'
 [main 92909de] Revert B, move A
 1 file changed, 2 deletions(-)

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
 \$ git lg1
 92909de (HEAD -> main) Revert B, move A
 3ce2309 C
 8370ac8 B
 d636259 A

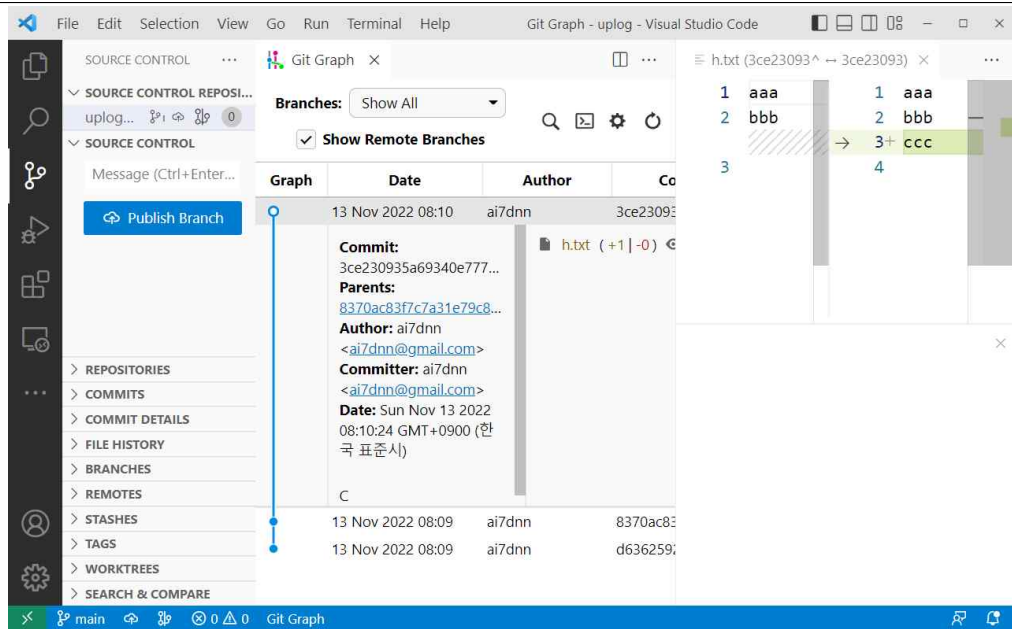
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
 \$ cat h.txt
 aaa

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/13w/uplog (main)
 \$ git reset --hard HEAD~
 HEAD is now at 3ce2309 C

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

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

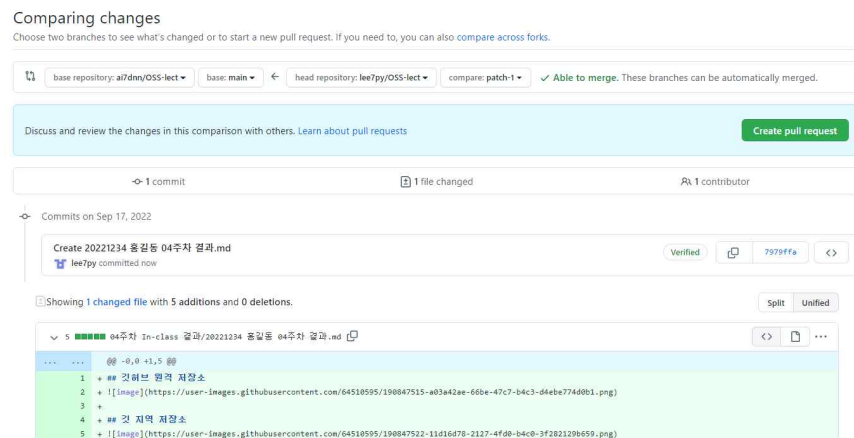
결과 PR로 보내기	<p># 다음 방법으로 PR(pull request) 해 주세요</p> <p><input type="checkbox"/> 팀장</p> <ul style="list-style-type: none"> • 팀 계정에서 지금 우리 수업의 Forked된 저장소를 새로 Sync한 후 <p><input type="checkbox"/> 팀원</p> <ul style="list-style-type: none"> • 자신의 계정에서 Forked된 팀 저장소를 새로 Sync한 후 • Forked된 팀 저장소의 이 폴더에서 다음 파일을 생성해 vscode 화면을 추가한 후 • vscode에서 Git graph와 커밋 C에서 파일의 차이를 열어 모두 보이게 화면 구성 ✓ QB-01-20221234-홍길동-13주차-결과.md
------------	---



- Forked된 팀 저장소에서 PR로 송부하고

□ 팀장

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



Open a pull request

