오늘배울내용

1. 특정한 Git commit을 삭제하거나 변경하기.

2. 리베이스 명령어 배우기

- 커밋메시지를 수정하거나 특정한 commit메시지를 제거하고자 할때 효과적으로 사용할 수 있다.

3. git commit 내용 지우기.

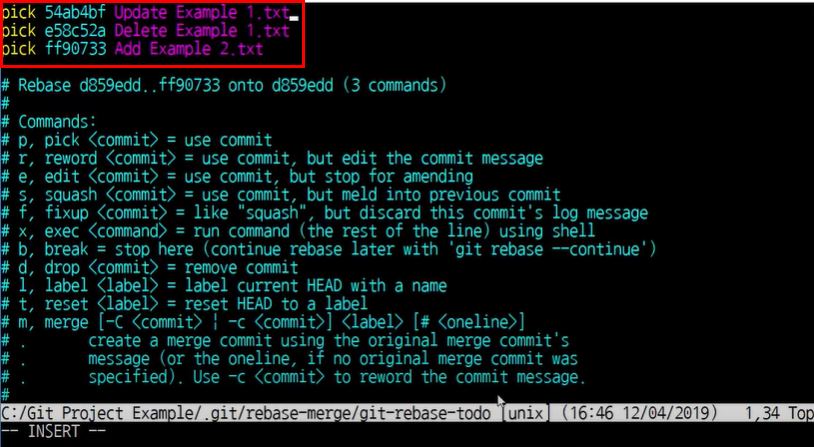
-------------------

Git rebase -I 를 하여 인터렉티브 모드로 들어가는 에디터가 열린다.

여기서 열린 인터렉티브 모드에서 어떤 커밋을 어떻게 수정할지 편하게 관리할 수 있다.

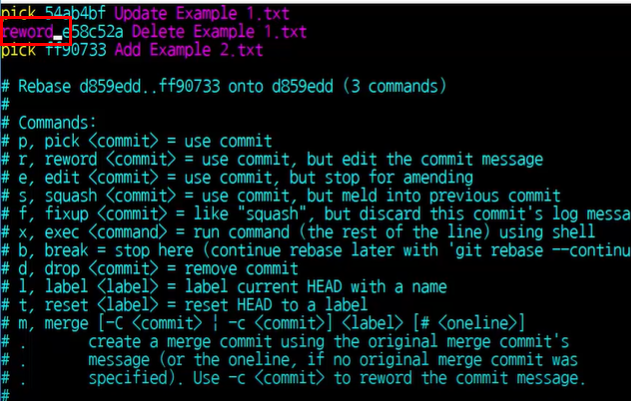


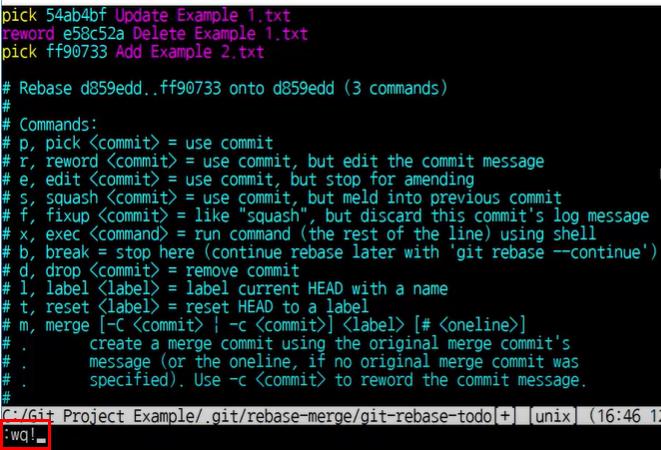
-I 뒤에 있는 HEAD~3은 머리부터 3개의 commit을 불러온다는 뜻이다.

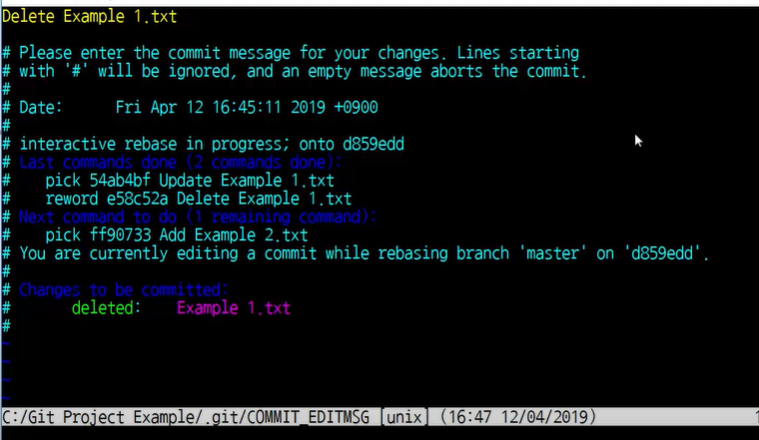


위에 빨간 상자는 최근 3개의 커밋 내용들이 나왔다.

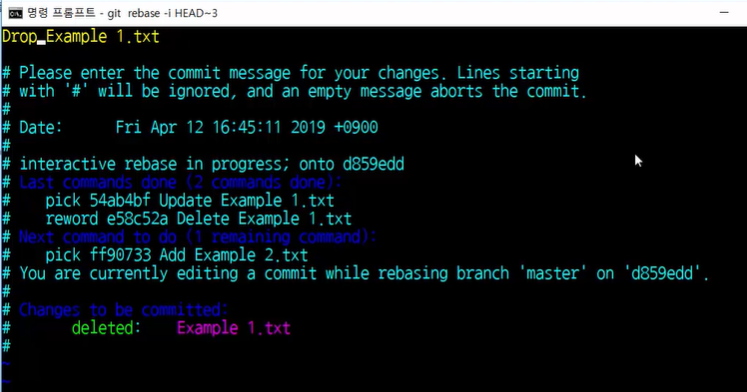
이중에서 2번째에 있는 Delete Example을 바꿔보겠다.

우선 pick 대신해서 reword를 써넣어주어 이름을 바꾸고 싶다고 써넣어 준다.

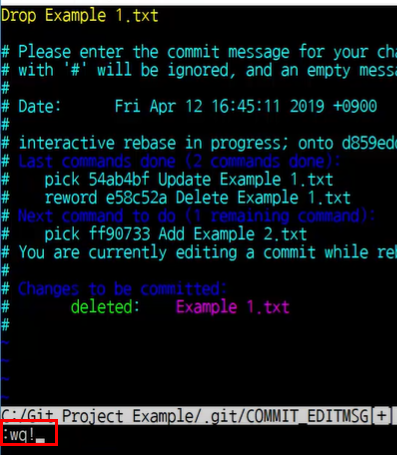
그리고 버튼 ESC를 눌러서 :wq!를 작성하여 해당 페이지를 빠져나온다.



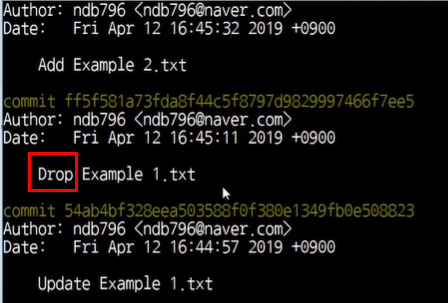
그러면 이렇게 어떤 곳을 고칠것인지 물어봐 주는데 여기서 자신이 원하는 데로 수정을 거치면 된다.



수정사항을 적은 뒤



아까와 마찬가지로 :wq!를 눌러서 해당 파일을 나와 주면 된다!



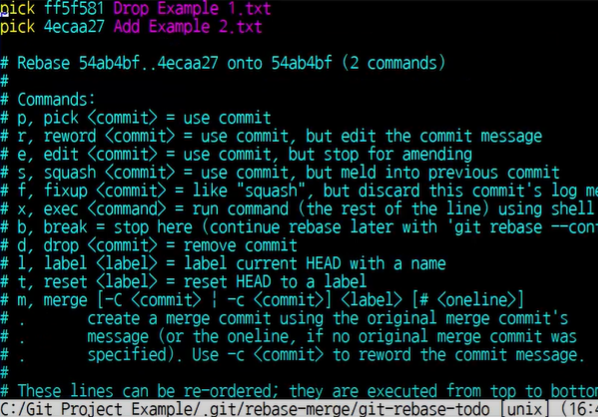
Git log를 사용하여 변경된 사항을 확인 해보면 이렇게 commit내용이 바뀐 것을 확인할 수 있다.

-------------------

2. 리베이스 명령어 배우기 +추가편



또한 HEAD~(숫자)가 아닌 바로 해쉬값을 넣어줄 수 있는데 이때 적용된 해쉬값의 위쪽 commit들이 보여지게된다.



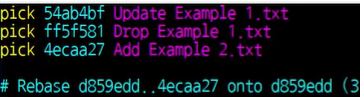
이렇게 해쉬값의 위쪽값이 표시된다.

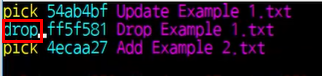
3. git commit 내용 지우기.

\*여기서 잠깐! 만약 혼자가 아닌 팀원들과 함께 이용하는 저장소라면 commit자체를 지우는 것은 권장되지 않는다.

(이유 : 충돌이 발생하기 때문이다.)



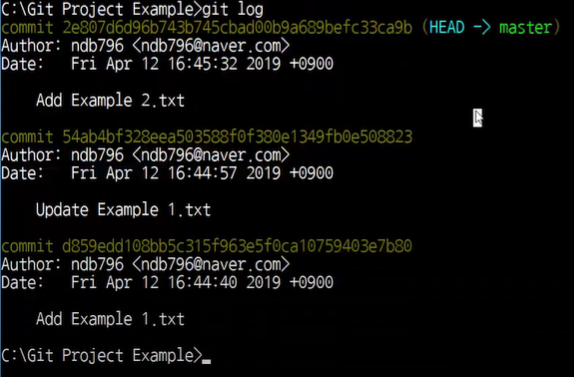




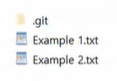
삭제를 할 수 있는 기능은 drop이기 때문에 drop을 이용하여 지워주도록한다.



프로그램을 나갈때는 반드시 :wq!를 사용하여 나가주어야한다.



Git log를 찍으면 원하던 commit이 사라져있는 것을 볼 수 가있다.



저장된 파일에 들어가 보아도 삭제한 commit의 활동이 사라졌다는 것을 알 수 있다.

(삭제한 커밋은 example 1.txt를 삭제한 커밋이였다. )

-즉 커밋을 삭제하면 해당 행동도 같이 사라진다.

-커밋을 삭제하면 그 특정 커밋이 마치 없었던것처럼 동작한다.

-git rebase는 다양한 기능이 있기 때문에 시간들여서 더 공부해보도록하겠다.