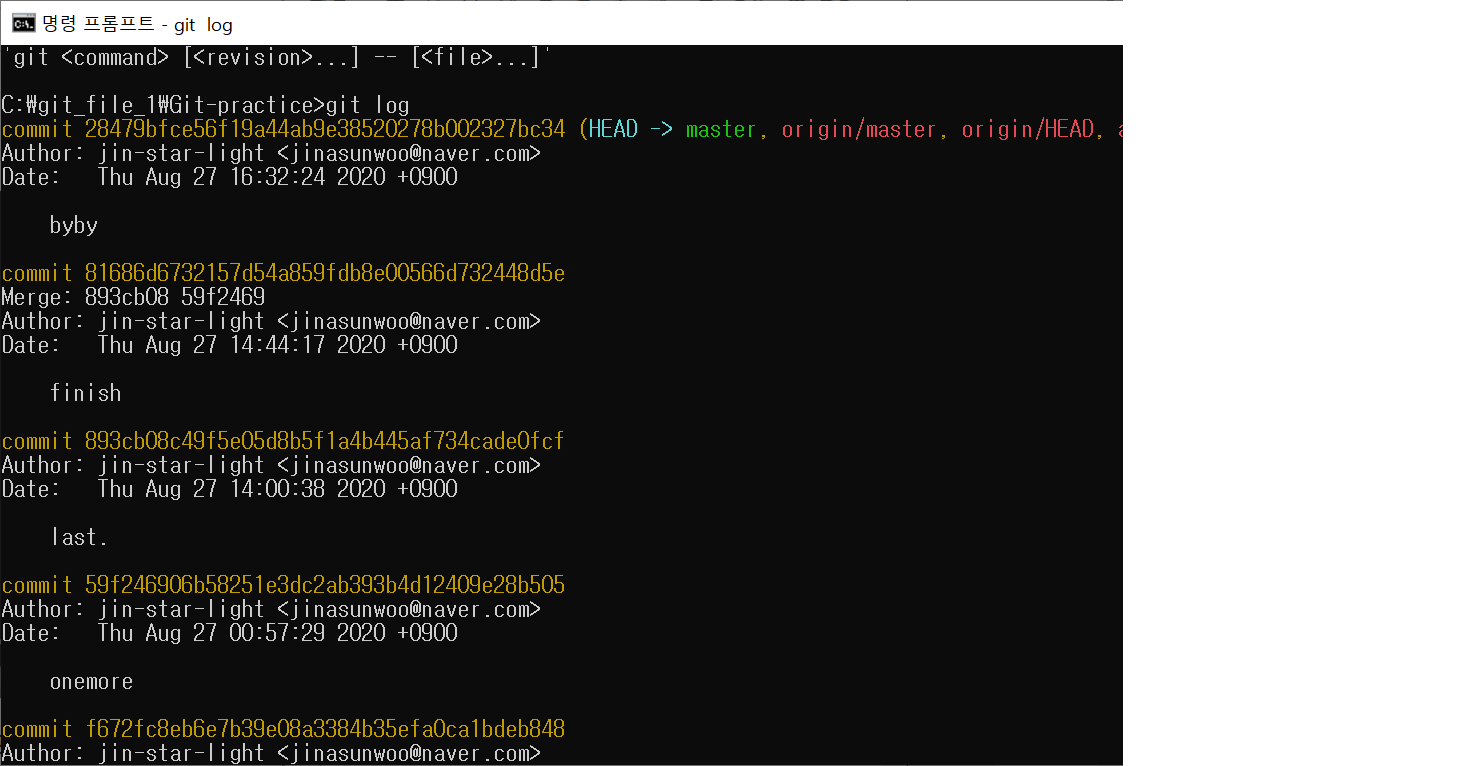
#오늘 배울 내용 #

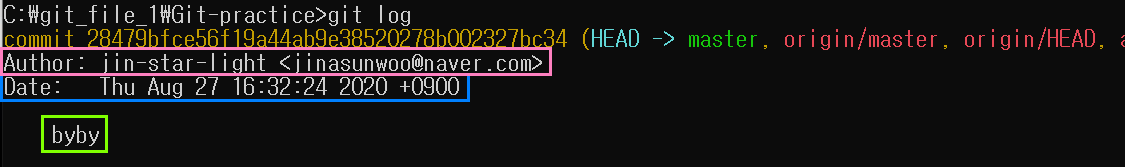
1. Git log에 대한 기본적인 정보들 .

2. git log에서 사용되는 추가적인 기능들

Git log 의 기능 : 처리내역을 한번에 볼 수 있다.



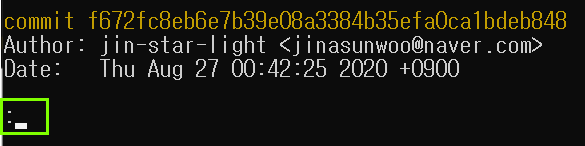
1. git log를 입력시 최신순서대로 변경된 파일 이력들을 확인할 수 있다.



2. [파란상자 ] 언제 커밋하였는지.

3. [핑크 상자] 누가 커밋하였는지

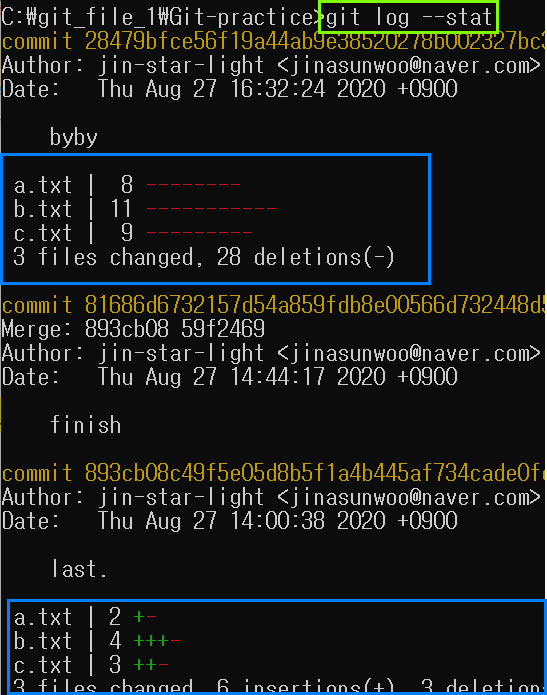
4. [초록상자] 어떤 이름으로 커밋하였는지 확인할 수 있다.



5. 너무많은 git log 내역이 있을 경우 위처럼 콜론으로 표시되며 엔터로 더 많은 log를 확인 가능하다.

6. q를 눌러서 git log를 그만 볼 수 있다.

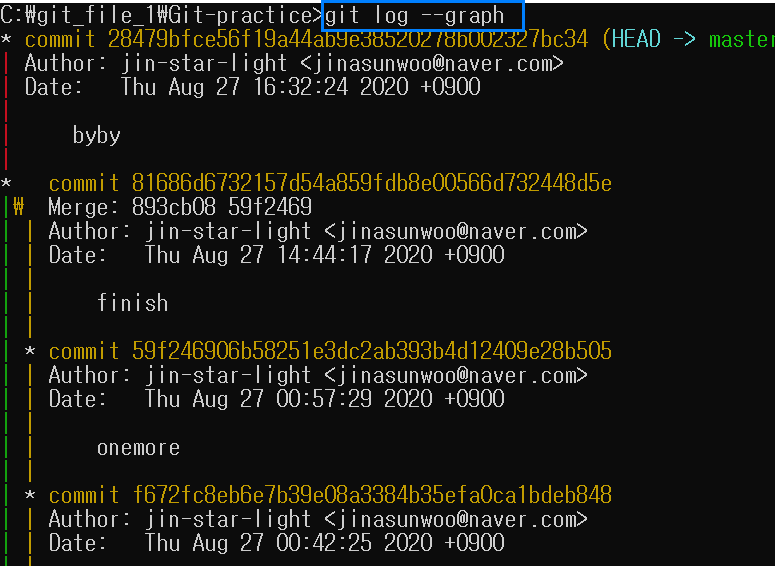
2. git log에서 사용되는 추가적인 기능들



Git log --stat 을 이용하여 각 커밋에 대한 통계정보를 볼 수있다.

파란상자에서 잘 보면 라인들이 어떻게 변경되었는지 시각적으로 보여준다.

-------



Git log --graph 는 브랜치의 병합정보를 그래프로 보여준다.



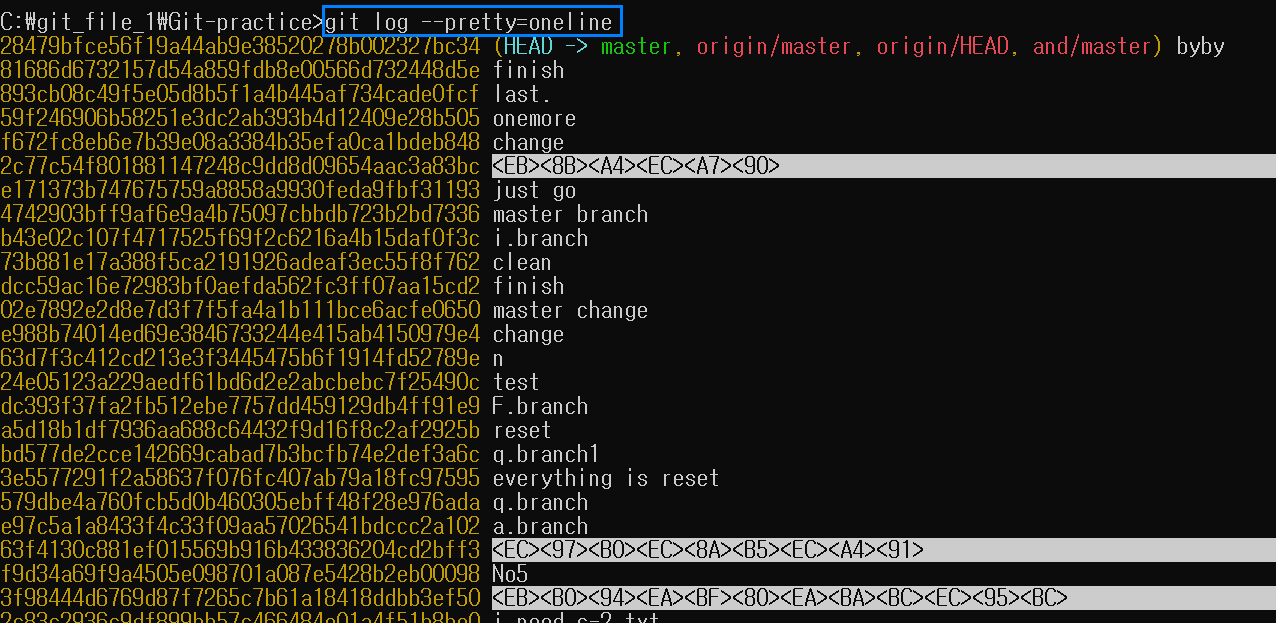
Git log -p를 사용해보았다.

Git log -p의 기능 :

지금까지 commit에 적용된 구체적인 항목들을 보여준다.

나는 git log -p -1을 사용하였는데 여기서 보이는 -1의 뜻은 가장 최근 것부터 1개 까지만 보여달라는 뜻으로 commit 한 개만 보여달라는 뜻이다.

-----------

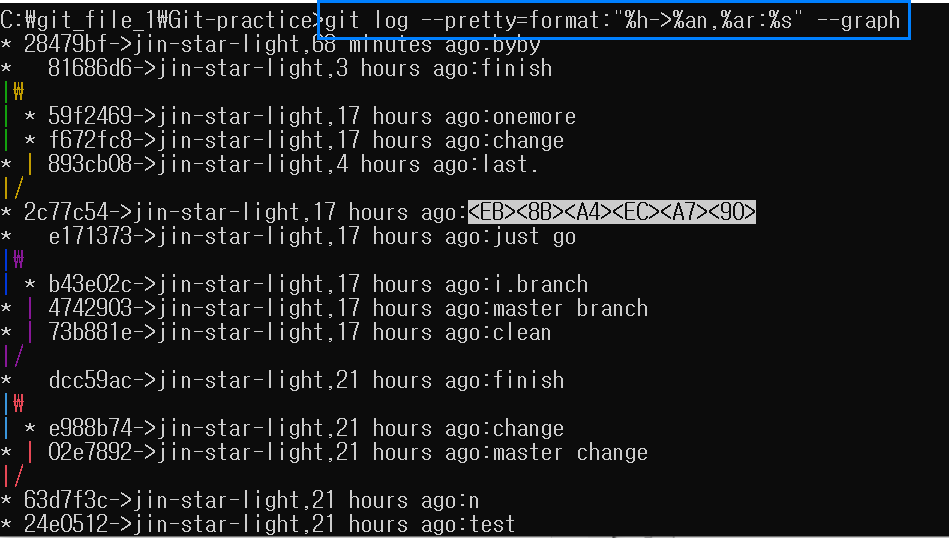


Git pretty : 우리가 지정한 형태로 commit을 보여주도록 한다.

Git log --pretty=oneline 을 사용하여 한줄로 commit을 보여주도록 나열하였다.

=oneline은 한줄로 나열하는 명령어로 commit이름과 해쉬값을 볼 수 있지만 생성된 날짜와 생성한 사람은 볼 수 없다.

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



이외에도 pretty와 함께 쓸수 있는 다양한 값들이 있다.

=format: 과 함께쓰는

%h : 커밋을 할때의 해쉬값을 출력해준다.

%an : 작성자 이름을 출력해준다.

%ar : 커밋된 날짜를 출력해준다.

%s : 커밋 주제를 출력한다.