교육 출처 : [Learn Git Branching](https://learngitbranching.js.org/?locale=ko)

**1-4 Rebase**

Rebase ?

🡪 리베이스는 기본적으로 커밋들을 모아서 복사한 뒤, 다른 곳에 떨궈 놓는 것.

아동 미술, 스크린샷, 핑크, 예술이(가) 표시된 사진

자동 생성된 설명

bugFix를 선택하고 bugfix 브랜치에서의 작업을 main 브랜치 위로 직접 옮겨 놓으려고 한다.지금 그림은 따로 개발했지만, rebase기능을 통해 순차적으로 개발한 것처럼 보인다.

스크린샷, 핑크이(가) 표시된 사진

자동 생성된 설명

스크린샷, 만화 영화이(가) 표시된 사진

자동 생성된 설명

다시 main브랜치 선택된 상황에서 bugFix로 리베이스 한다면,

스크린샷, 원이(가) 표시된 사진

자동 생성된 설명

Main이 bugFix보다 부모 브랜치에 있었기에 단순히 브랜치 이동만 하면 된다.

2-1 Head 분리하기

깃에서 Head란 ?

🡪 Head는 현재 체크아웃된 커밋을 가리킨다.

다시 말해 현재 작업중인 커밋이다. Head는 항상 작업트리의 가장 최근 커밋을 가리킨다. 작업트리에 변화를 주는 git 명령어들은 대부분 head를 변경하는 것으로 시작한다. Head는 브랜치의 이름을 가리키고 커밋을 하게 되면 bugFix의 상태가 바뀌고 이 변경은 head를 통해 확인이 가능하다.

스크린샷, 핑크, 하트, 그래픽이(가) 표시된 사진

자동 생성된 설명

여기서 c2 브랜치를 만들고 head를 c2로 옮겨보자.

스크린샷, 다채로움이(가) 표시된 사진

자동 생성된 설명

전에 배웠던 checkout 명령어를 통해 head의 이동이 가능하다.

스크린샷, 핑크이(가) 표시된 사진

자동 생성된 설명

여기서 c1으로 head를 이동시킨다면?

스크린샷, 원, 핑크이(가) 표시된 사진

자동 생성된 설명

C1 브랜치로 head가 이동되는 모습을 볼 수 있다.

**2-2 상대참조1)**

git에서 여기저기 이동할 때 커밋의 해시 사용하는 방법은 복잡하다. 실제 깃에서 값을 확인해보면 “fed2da64c0efc5293610bdd892f82a58e8cbc5d8” 엄청 복잡한 걸 확인할 수 있다. git에선 이 해시 값 대신 fed2만 입력하면 해결된다.

하지만, 커밋들을 해시로 구분하고 사용하는건 다소 불편하다고 봐야한다. 여기서 git의 상대 참조(Relative Ref)가 등장한다.

상대 참조로 우리가 기억할 만한 지점에서 출발해서 이동해 다른 지점에 도달해 작업 할 수 있다.두 가지 기능이 있는데,

\*\* 한 번에 한 커밋 위로 움직이는 ^

\*\* 한번에 여러 커밋 위로 올라가는 ~<num>

스크린샷, 그래픽, 하트, 다채로움이(가) 표시된 사진

자동 생성된 설명

다음 그림에서 main의 부모를 체크아웃 해보자. (git checkout main^)

스크린샷, 다채로움, 핑크이(가) 표시된 사진

자동 생성된 설명

2-3 상대 참조 2)

커밋트리에서 위로 여러 단계 올라가고 싶을 때 ^ 이 있는데 이 외에 더 좋은 틸드 (~) 연산자가 있다. 틸드 연산자는 선택적으로 올라가고 싶은 부모의 개수가 뒤에 숫자가 온다.

핑크, 스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷, 원, 디자인이(가) 표시된 사진

자동 생성된 설명

돌아가고 싶은 커밋 개수를 ~ 뒤에 숫자로 명시한다. (git checkout HEAD~4)

브랜치를 강제로 옮길 방법이 있는데, -f 옵션을 이용해 브랜치를 특정 커밋에 직접적으로 재지정 할 수 있다.

스크린샷, 텍스트, 레드, 카민이(가) 표시된 사진

자동 생성된 설명

다음 main 브랜치를 head로부터 3칸 떨어지게 해보자. (git branch -f main HEAD~3

)

스크린샷, 그래픽, 만화 영화이(가) 표시된 사진

자동 생성된 설명

2-4) Git에서 작업 되돌리기

Git에서 작업한 것을 되돌리는 여러가지 방법이 있다.

하나는 git reset, 다른 하나는 git revert를 사용하는 방법이다.

핑크, 하트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

다음 그림에서 C2를 삭제하고 HEAD를 C1으로 되돌아가보자. (git reset HEAD~1)

핑크, 스크린샷, 하트이(가) 표시된 사진

자동 생성된 설명

각자의 컴퓨터에서 작업하는 로컬 브랜치의 경우 리셋을 잘 쓸 수 있지만, “히스토리를 고쳐쓴다”는 점 때문에 사람이 작업하는 리모트 브랜치에는 쓸 수 없다. 변경분을 되돌리고, 이 되돌린 내용을 다른 사람들과 공유하기 위해선 git revert를 사용한다.

핑크, 스크린샷, 하트이(가) 표시된 사진

자동 생성된 설명

C2 브랜치와 HEAD의 복사본을 만들어보자.

스크린샷, 핑크이(가) 표시된 사진

자동 생성된 설명

질문- main은 브랜치인가?