

g01

.git 디렉토리에는 저장소에 필요한 뼈대 파일이 들어있다.  
프로젝트에 필요한 모든 정보와 커밋  
원격 리포지토리 주소 ( 깃허브 주소 ) 등 정보가 포함되어 있다.  
또한 커밋기록을 저장하는 로그도 포함되어 있다.  
그러면 우리가 뭘 커밋한지도 나와 있을 것 같다.  
내가 설정한 이름의 브랜치이름? 그것도 쓰여있는데

오 커밋할 때 올린 메시지도 볼 수 있다.

.git 의 config는 일단 나의 깃헙 주소가 적혀있고 어디로 merge 하는지  
등등이 적혀 있는 것 같다. 위의 core는 대설정? 전체 설정 비슷한거 같다.  
음 그리고 logs>refs>heads 는 내가 이때까지 커밋한 내용과 msg가 적혀 있는거 같다.

objects 파일은 뭘까..

뭔가 16진수로 설정된 2글자 폴더도 있다. 안에는 밑과 비슷한 40자 파일인데 열어보니 암호  
화되어 있는건지 읽을 수 없다..

xT+)JMU040g040031Qrut?瑯Mapd???뽕뽕??幸]QJPE뽕)z??땡OX뽕?{實뽕?뽕?R?S綾]\*J뽕}4{膽  
k뽕?胤뽕z?< 韞,?

>> Git은 모든 것을 해시로 식별해 쓰기에 이런 문자열은 여기저기서 보입니다. 저장소에 보  
관하는 파일도 이 체크섬을 키로 해서 보관하고, 전체 트리(디렉터리)도 이 체크섬으로 보관하  
며, 커밋에 대한 데이터도 이 체크섬 기준으로 보관합니다.

그리고 찾고자 하는 범위를 “한 저장소 안”으로 좁히면, 앞의 몇 글자만 써도 대부분 달리 겹  
치지 않아서, 여덟 자나 열 자로만 줄여서 쓸 수도 있습니다. : 그러면 이거는 한 저장소 안에  
서 어떤 것이 어느 폴더에 드갈지 알려주는 키 인가?

g02

브랜치는 우리가 계속 마스터에 pull을 하면 버그가 있거나 큰 오류가 있는 것을 마스터에  
pull 해버리는 경우가 생기기 때문에  
브랜치와 서로 영향을 받지 않는  
임시 파일? 프로토타입을 생성해 테스트해보는 그런 역할로 쓰인다.

브랜치는 독립적으로 어떤 작업을 진행하기 위한 개념이다.  
동시에 여러작업을 하면 뭐가 바뀌었는지 마스터에 바로바로 확인하기 어렵기 때문에

뭐 기능추가브랜치, 버그수정브랜치 등으로 나눠서 잘 작동하는지, 잘 작성됐는지 확인하고  
마스터에 병합해서 마스터가 중구난방이 되지 않도록 막는 역할? 기능인 것 같다.

g03 fork clone

fork는 다른 사람의 Github repository에서 내가 어떤 부분을 수정하거나 추가 기능을 넣고  
싶을 때 해당 repository를 내 Github repository로 그대로 복제하는 기능이다. fork한 저장소  
는 원본(다른 사람의 github repository)와 연결되어 있다. 여기서 연결 되어 있다는 의미는  
original repository에 어떤 변화가 생기면(새로운 commit) 이는 그대로 forked된  
repository로 반영할 수 있다. 이 때 fetch나 rebase의 과정이 필요하다.

(공짜 다른 사람 레포지토리를 그대로 내 레포지토리에 복사해오는 것? 그리고 다른 사람 레  
포지토리에 내가 한 것을 주고싶다? 커밋하고 싶으면 pull request하면은 그 사람이 승인 하  
면 바로 pull 되는기고 )

clone은 특정 리포지토리를 내 컴퓨터(아니면 다른 디바이스)로 복사해 저장하는 것  
또한 기존의 제일 처음 original repository와 연결되지 못한다. 즉 저장소의 commit, 등의  
로그를 보지 못함(이게 큰 단점인가?)

중요한 점은 a와 b가 클론 해서 작업을 하다 b가 pull하면 a는 그 pull한 것을 fetch 해야  
pull 할 수 있다.