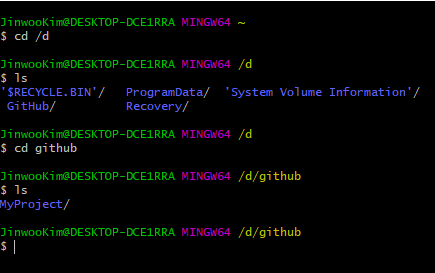
깃 bash 기준

따라가기 전에, 우리 harmony깃허브에 적용시키기 전에 먼저 test repository를 만들어서 진행해보길 바라

먼저, 저장 폴더 설정을 해줘야 하는데 나중에 데이터가 많은 프로젝트를 진행하면 저장공간이 많이 필요한데 다시 설정하기 귀찮으니까 저장공간이 여유가 있으면 C드라이브에 없으면 D드라이브에 미리 깃 허브 폴더를 지정하는게 좋아

난 참고로 이런 식으로 지정했어. bash기준

D드라이브에 Github라는 이름의 저장 폴더 만들어놓은거야.

난 잘 모를 때 해서 MyProject라는 쓸데없는 폴더를 한번 더 만들었으니 같은 실수하지말길

이제 해야할 부분은 깃에 너가 누군지를 알려줄 부분



참고로 너가 써야할 이메일은 너가 깃 가입할 때 쓴 이메일이야. 이름은 노상관

다음은 해당 디렉토리를 로컬 깃 저장소로 확인시켜주는 단계야



나는 이미 initialize를 한번 했기 때문에 이렇게 나오지만 너넨 다른 문장 나올거야

로컬 저장소를 init 했으면 이제 로컬저장소와 원격저장소를 연결해야 하는데



이런 식으로 git remote add origin 깃저장소주소.git 형태로 명령창에 적어주면 돼

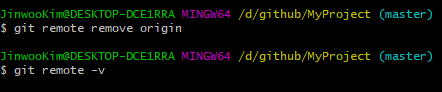


git remote -v 를 치면 연결 상태를 확인할 수 있어

참고로 나중에 너가 이 프로젝트 말고 다른 프로젝트를 깃과 연결해야할텐데 그럴 때 기존 repository remote를 제거해야 해



이거 이후에 연결 상태 확인하면 아무것도 안뜬 것을 볼 수 있엉



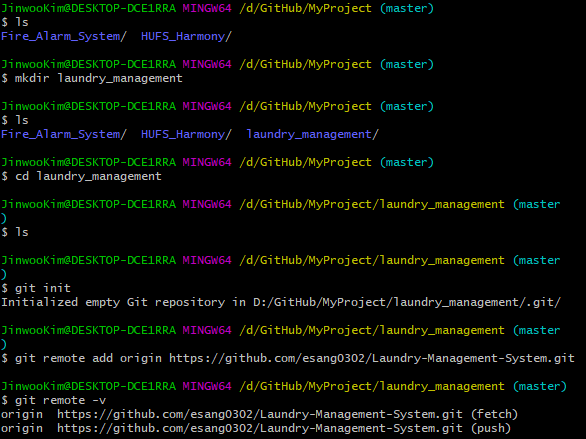
여기까지 연결 끝.

기본적으로 깃은 push 와 pull로 나뉘는데,

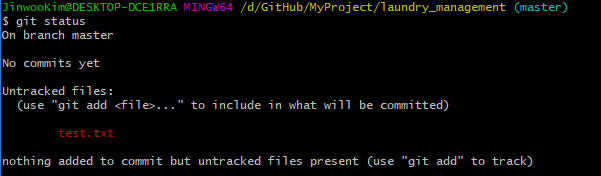
git push 명령어는 파일을 upload되는데 remote repository의 master branch에 upload된다고 보면되고 git pull은 반대로 download받는다고 생각하면돼

내가 한걸 따라오는게 빠르니까 예시를 보여줄게

일단 나는 laundry\_management라는 디렉토리를 생성했어 (np때 했던 자료를 github에 올리는걸 예시로 들게)



폴더(디렉토리)는 만들었지만 비어있지? 테스트로 test.txt를 해당 폴더에 넣어볼게



git status 라는 명령어를 통해 해당 remote repository에 어떤 변화가 생겼는지를 볼 수 있어

캡쳐 화면과 같이 add 되어있지도 commit도 아직 안되어있는 빨간 글씨로 뭔가 추가된게 보일거야

(들어가기 전 정리: committed: 파일을 수정한 후 해당 파일에 대해 commit 명령을 실행해 파일을 로컬 레퍼지토리로 안전하게 저장한 상태

Modified: 파일을 수정한 후 아직 로컬 레퍼지토리에 commit하지 않은 상태

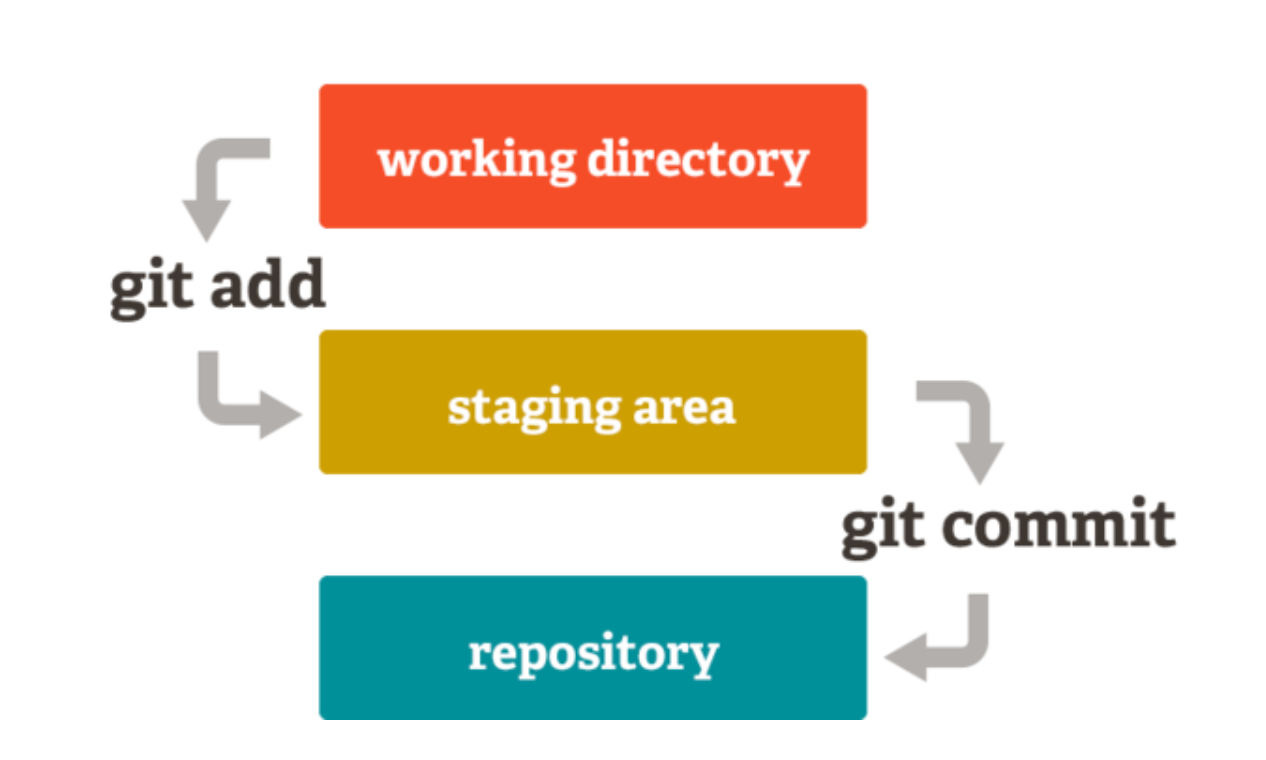
Staged: 파일을 수정한 후 수정할 파일을 곧 commit할 것이라고 표시한 상태)

본인이 편집한 파일을 레퍼지토리에 올리려면 기본적으로 add -> commit -> push의 단계를 거쳐야해.

git add . : 현재 디렉토리의 모든 파일을 track 하기위해 추가

git add 파일명.확장자 : 선택적으로 추가

git commit -m “동사를 현재형으로 설명 comment남기기 (무슨 작업? 무슨 편집? 을 했는지”)



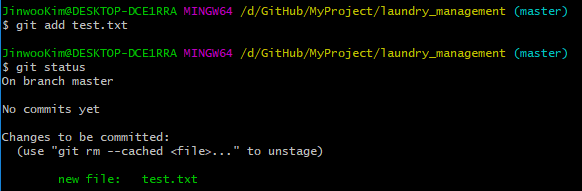
Working directory:

Project의 특정 branch를 checkout한 내용

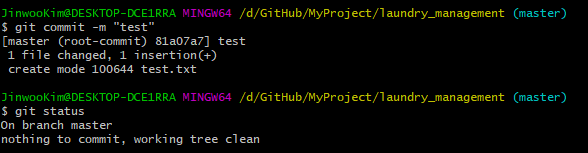
Staging area:

Git directory에 존재하는 단순한 파일. 곧 commit할 파일에 대한 정보를 가지고 있다.

Repository(git directory): Git이 프로젝트의객체 데이터베이스를 저장하는곳. 특정 폴더를 local repository로 설정하려면 git init 명령 이용.

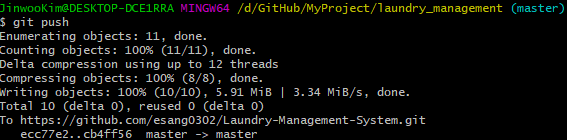


우선 git add를 한 실행창이야. 상태를 살펴보면 committed되기위해 바뀐 파일이 무엇인지 볼 수 있을거야. 이 파일을 commit 하면 돼



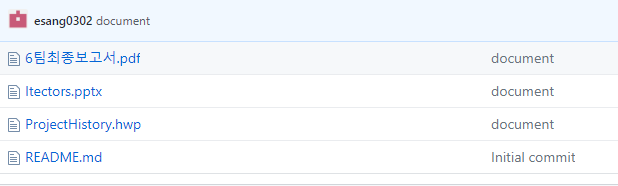
git commit 한 화면이야. 읽어보면 1 insertion(+) 된 게 나오지? 너가 파일 하나 추가했다는 뜻~

이제 git push 해주면 끝 (안되면 git push origin master)



중간에 파일 올린거 다른거로 바꿔서 헷갈리수도 있겠지만….

이후에 github 들어가서 제대로 들어갔나 확인해봐



짜잔

이제 너가 작업해 놓은 파일들을 harmony repository에 넣을건데

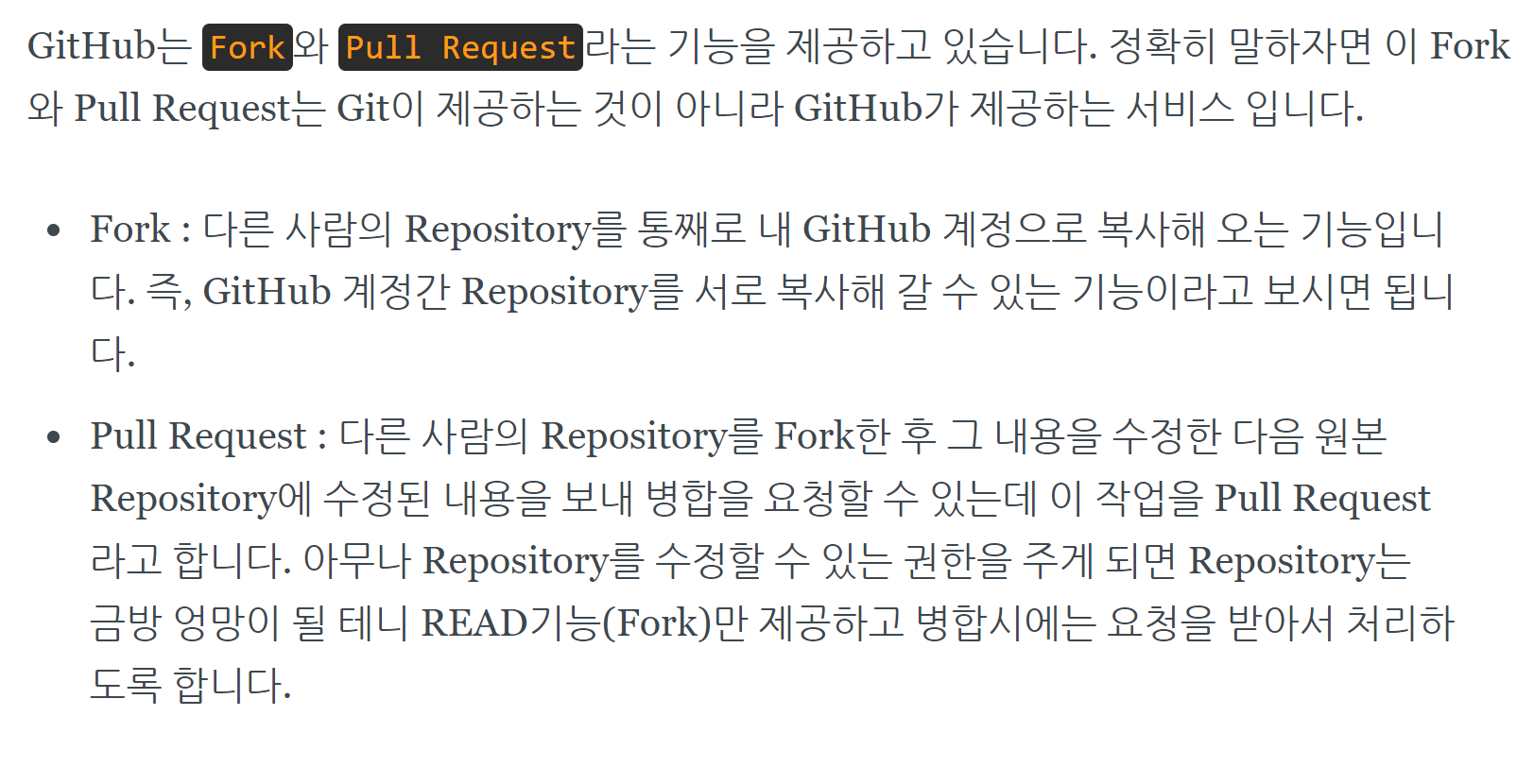
여기서 제일 중요한건 우리가 branch라는 개념을 안쓰기 때문에 충돌을 방지할 방법이 서로에게 깃을 쓰고있느냐고 물어보는 수밖에 없으니 깃에 무얼 올리거나 받기전에 팀원들에게 물어볼 것

그리구 push전에 항상 pull이 이루어져야할 것

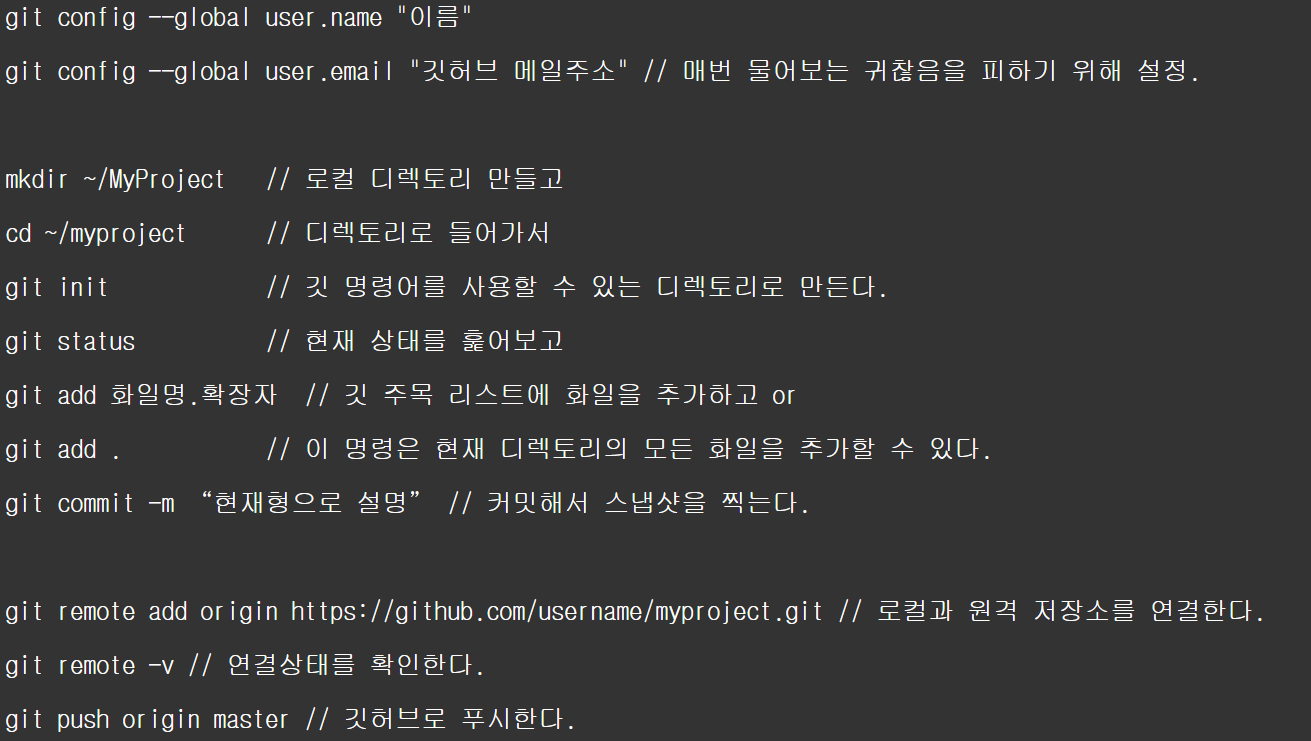
만약에 너가 pull하기전에 push를 한다면 너가 깃을 안쓰는동안 팀원들이 깃을 업데이트 해놓은 정보가 씹히고 너의 컴퓨터에 있는 정보만 push되면 다른 사람들이 작업해서 깃에 올린게 다 날아가는거야~ 물론 각자 컴퓨터엔 작업했던게 저장되어 있겠지만서도 그전까지 누가 마지막으로 작업했는지 역추적해야하니까 그런일이 발생하면 싸우자는걸로 간주할게 ^\_^

#참고 (내 설명이 부족하니까)



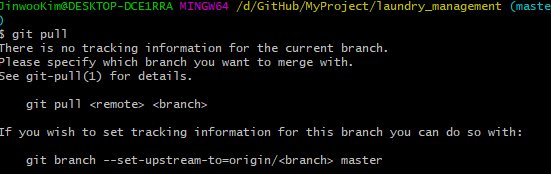


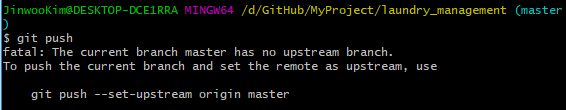
참고 2)





참고 사)





이런 오류 날 때

