Github

—————————————————————————————

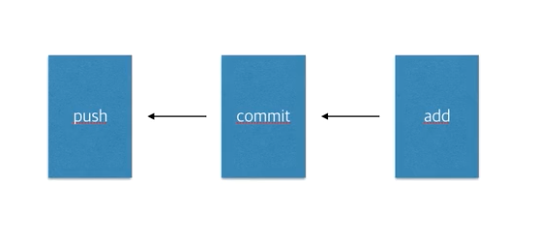
\*위키피디아 같은 것. 코딩한것을 메일같은것으로 보내는 게 아니고 인터넷저장소인 github에 저장하고 다른 사람들이 접근해서 공동작업하기 쉽게 한다.

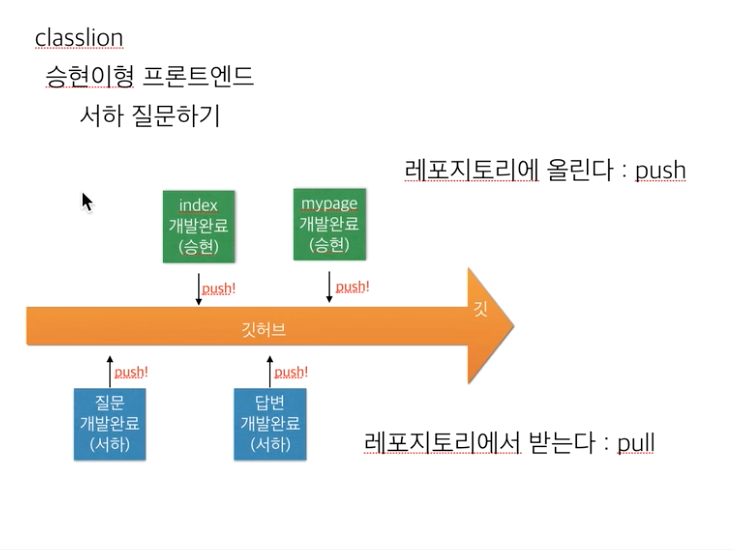
\*순서

1)add : 변경한 파일 중에 저장할 것을 지정하는 기능

2)commit : 지정한 파일들을 묶에서 내컴퓨터에 저장

3)push : 내컴퓨터에서 깃허브로 보내는것



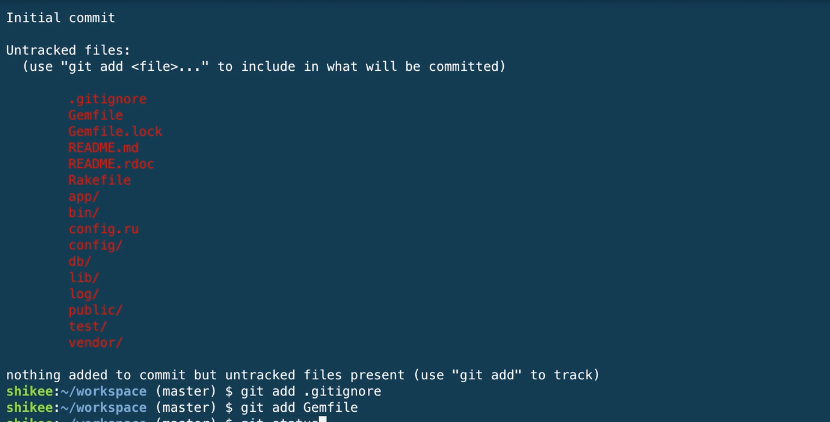


\*프로젝트에 깃을 넣기(프로젝트에 깃 저장소를 만들기) : git init 입력

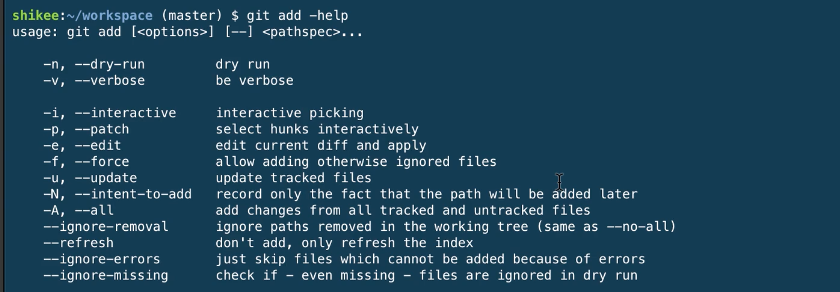
\*깃 상태확인 후 add

git status 입력하면 아래와 같이 나오고

그후 commit할것을 add 하기 위해서 저장할 파일 지정한 후 git add Gemfile와 같이 입력(모든 파일을 add 할때는 git add - - a 작대기두개a)



cf) git help 치면 도움말 나옴, add에 대한 도움말은 git add -help라고 침



아래 그림처럼 git commit 으로 저장가능하고 -m “메시지”를 통해 작업한것에 대한 설명을 넣어준다.(메세지를 반드시 넣어줄것! 다른사람도 알수있게)

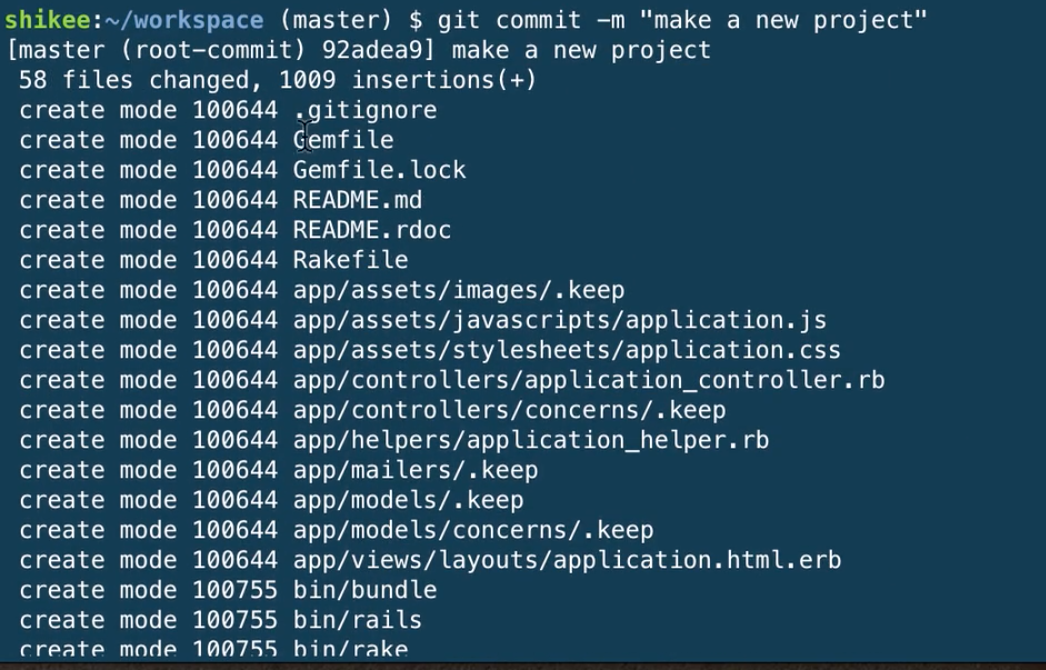
줄 해석)

58 files changed, 1009 insertions(+):58파일 바뀌고, 1009인설션있었다.

master(root-commit) : 첫번째 저장소는 master라는 이름이다.

92adea9 : 커밋의 이름

make a new project : 메세지



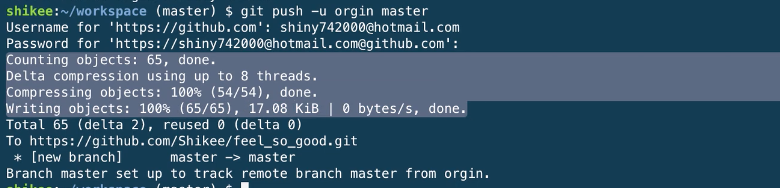
아래의 화면은 guthub를 만들었을때 나오는 화면이다. 두번째 그림에서 git init, git add,commit은 위에서 다 했으니 remote를 해주면 된다.

대부분 첫번째 저장소 이름은 origin으로 하니 저걸로 사용하면 된다. 그리고 주소는 아래그림의 첫번째 그림에서 나온 주소를 입력한다.

마지막 줄 push -u에서 -u 가 어떤의미인지 알려면 git push -help입력하고 찾아본다(첫번째로 할때는 -u로서 set upstream을 해주는 의미)



그리고 아래처럼 적어주면 username과 password물어 보는데 그것까지 적어주면 완료된다.



그 후에 파일이 변경되었다면

1)git status로 확인 해보고 2)git add —a등으로 add

3)git commit -m “add controller n1”

4)git push origin master