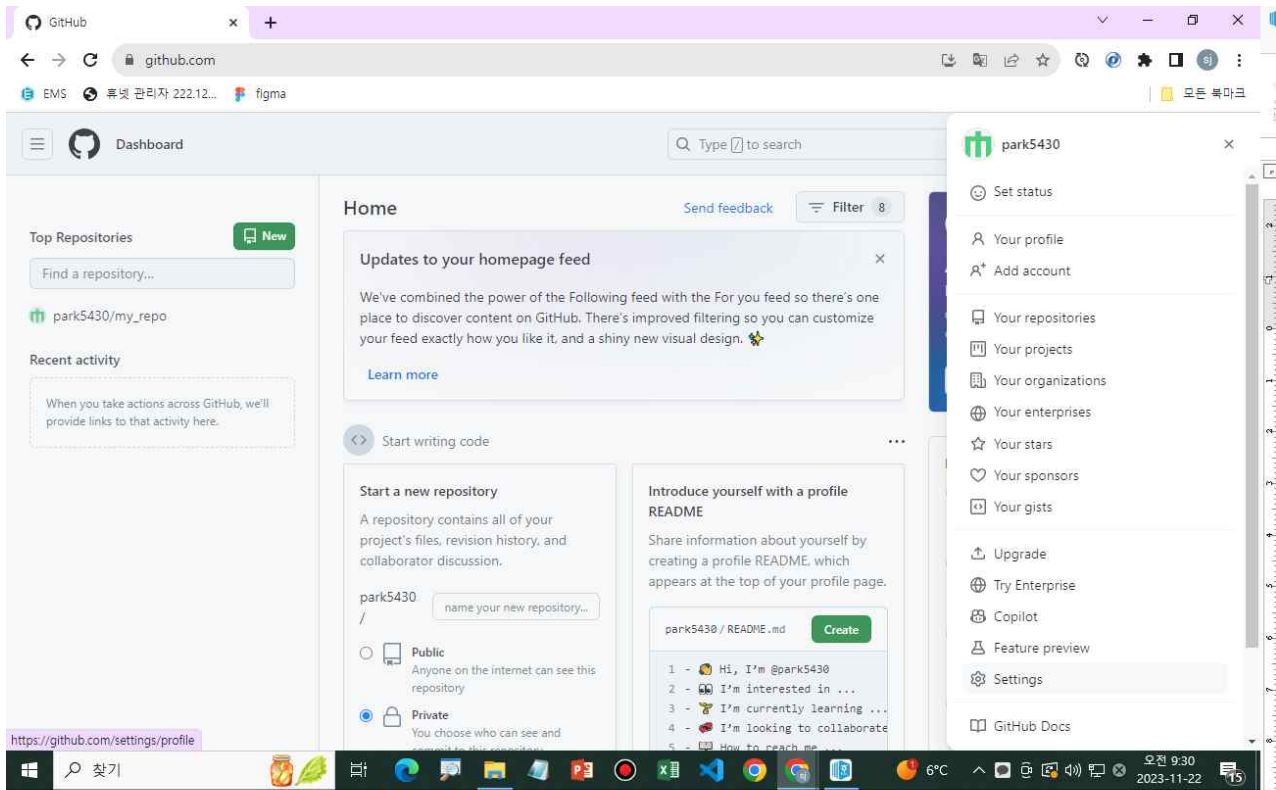


git 사용 총정리

1. 로그인 해서 저장소를 생성한다. (이 때, 개별적인 파일 업로드 히스토리가 있는 경우, git init 생성 이전의 히스토리에 대한 정보 관리를 요구한다. 그래서 시스템이 git 기능을 제대로 수행하지 않는다.
2. ***git 사용 관련해서는 새로 repo만든 이후, 개별적으로 파일 올리지 말 것
3. 지금 유형의 문제는 웬만한 블로그에서 다루지 않는다.
4. 또다른 문제는 git 지정 저장소에 하위 저장소 등은 함부로 날리지 않는다. 같은이유로 히스토리 관리 에러가 발생하니까...



로컬 컴퓨터 내부 원하는 폴더를 선택해서 코딩내역을 github에 보낼 전송폴더로 등록한다.
혹시나 위와같은 이전 히스토리 문제 발생 시, git폴더를 전부 지우고 다시 만든다.

*** 공용 컴퓨터는 c드라이브 접속을 보안문제로 막아둔다. 그래서 다운로드폴더 내 하위폴더 같이 모두가 통상적으로 사용하는 곳에 설정해야한다.

1. 폴더선택 후 마우스 우클릭으로 open git bash 선택
2. \$ git init 치면 해당폴더 등록이 초기화 된다. + 폴더보기에서 숨김파일 표시로 설정 .git 숨김폴더가 생성된다. git에서는 컨트롤 c 이런 단축키 안먹히니까 오른쪽 마우스로 대신한다.
3. 내 계정을 등록한다. 계정등록은 (--replace-all 은 기존 공용계정 등록 아이디 밀고 다시쓰기)

```
git config --global --replace-all user.email "sangjpar@gmail.com"
```

```
git config --global --replace-all user.name "park5430"
```

내 계정이 내 로컬 컴퓨터에 등록되었다.

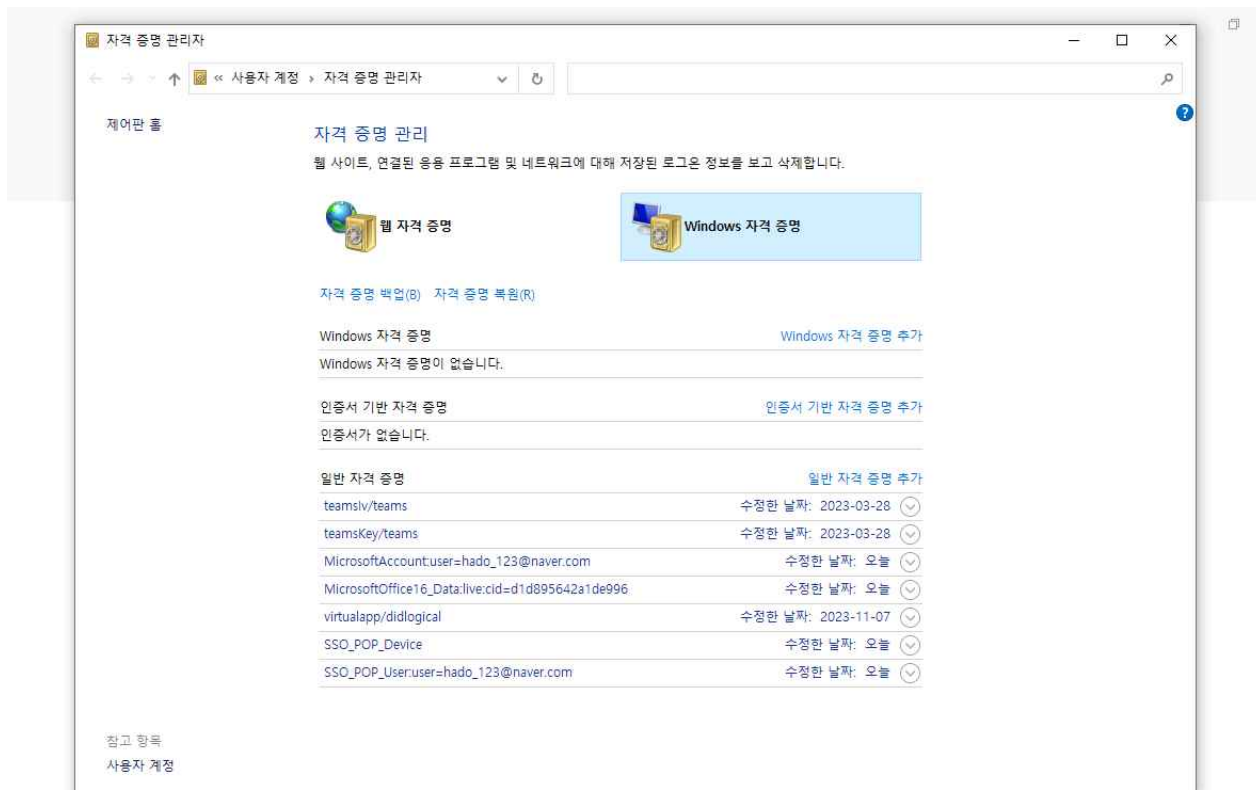
공용컴퓨터의 또다른 문제는 사전에 다른사람이 자격증명 관리자에 아이디, 비밀번호를 설정하는 경우가 있다. 일반 자격증명 항목에서 해당 구 유저를 없애야 한다.

문제발생 시, 아래 블로그 참조하여 조치

<https://yunseul-light.blogspot.com/2017/08/github-windows-git.html>

자격증명 관리자 제어판에서 검색 후 windows 자격증명으로 들감.

들가서 아래보면 다른유저 계정이 선 등록된 경우가 있는데 그래서 계정충돌이 나는 거였음.



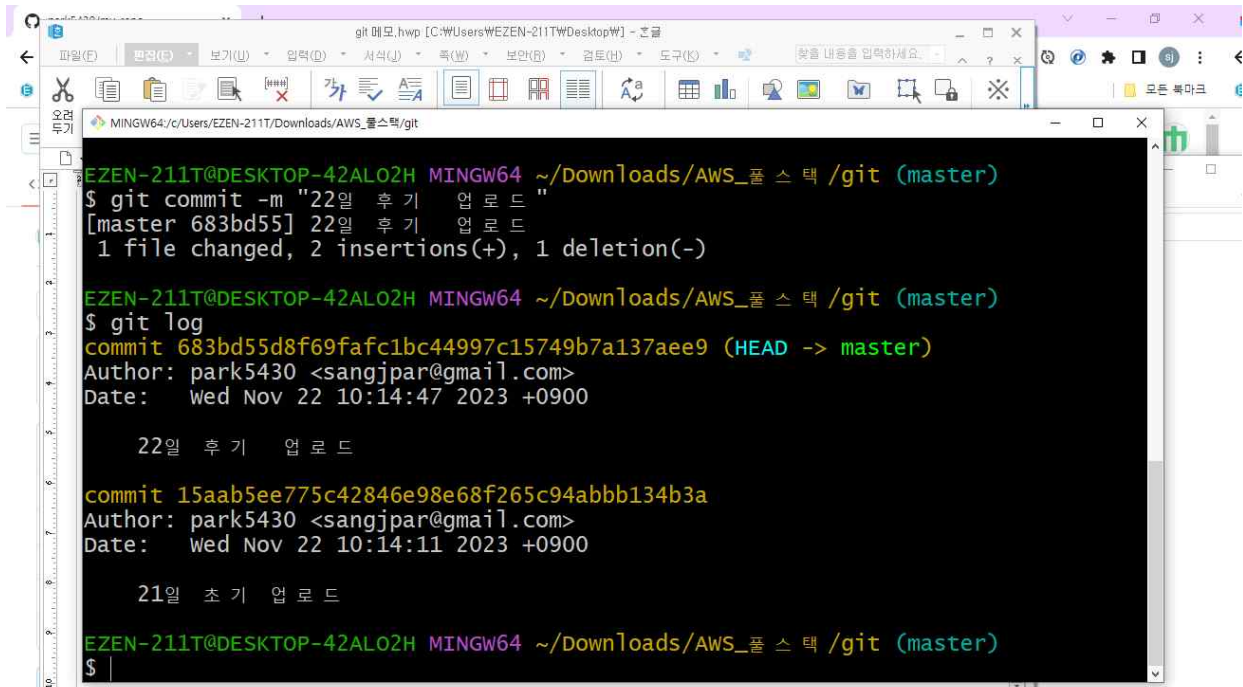
그다음에 내가 작업할 파일을 생성해서

4. 작업파일 등록: `git add 내파일이름`

5. 작업파일에 대한 설명 등록: `git commit -m "관련설명 추가"`

6. 원하는 만큼 파일수정, 재업로드, 설명 추가등록 하기.

7. 내가 작업한 모든 파일 히스토리는 `git log`라고 치면 나옴



```
EZEN-211T@DESKTOP-42ALO2H MINGW64 ~/Downloads/AWS_폴 스택 /git (master)
$ git commit -m "22일 후기 업로드"
[master 683bd55] 22일 후기 업로드
1 file changed, 2 insertions(+), 1 deletion(-)

EZEN-211T@DESKTOP-42ALO2H MINGW64 ~/Downloads/AWS_폴 스택 /git (master)
$ git log
commit 683bd55d8f69fafc1bc44997c15749b7a137aee9 (HEAD -> master)
Author: park5430 <sangjpar@gmail.com>
Date: Wed Nov 22 10:14:47 2023 +0900

    22일 후기 업로드

commit 15aab5ee775c42846e98e68f265c94abbb134b3a
Author: park5430 <sangjpar@gmail.com>
Date: Wed Nov 22 10:14:11 2023 +0900

    21일 초기 업로드

EZEN-211T@DESKTOP-42ALO2H MINGW64 ~/Downloads/AWS_폴 스택 /git (master)
$
```

8. 원하는 버전의 파일로 원복: `git checkout 683bd55`

이렇게 앞 7자리를 입력한다. 그러면 해당 버전으로 파일이 원복된다.

`git checkout -`

하면 가장 최근 버전의 파일로 원복된다.

9. 내 계정의 저장소 등록

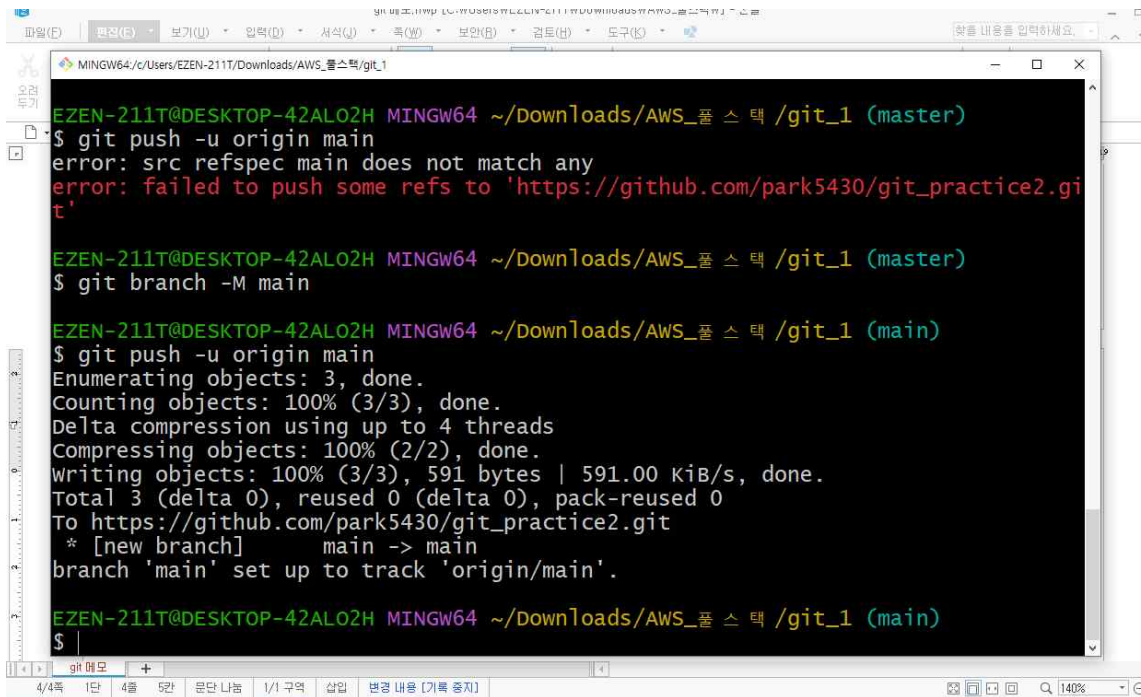
`git remote add origin https://github.com/park5430/git_practice2.git`

10. 저장소 안에 main 하위 브랜치 등록

`git branch -M main` (main 브랜치에 안넣을거면 다른 만들어진 브랜치 이름 등록하기)

11. 로컬 컴퓨터 내 add & commit으로 히스토리 올라간 것을 내 github에 업로드 하기

git push origin main (역시 main 브랜치 아니면 원하는 브랜치 이름 입력)

A screenshot of a Windows terminal window with a black background and green and white text. The prompt is 'EZEN-211T@DESKTOP-42ALO2H MINGW64 ~\Downloads\AWS_폴 스택\git_1 (master)'. The user enters '\$ git push -u origin main', which results in an error: 'error: src refspec main does not match any' and 'error: failed to push some refs to \'https://github.com/park5430/git_practice2.git\''. The user then enters '\$ git branch -M main', changing the branch to 'main'. The prompt now shows '(main)'. The user enters '\$ git push -u origin main', which successfully pushes the code to the remote repository. The output shows 'Enumerating objects: 3, done.', 'Counting objects: 100% (3/3), done.', 'Delta compression using up to 4 threads', 'Compressing objects: 100% (2/2), done.', 'Writing objects: 100% (3/3), 591 bytes | 591.00 KiB/s, done.', 'Total 3 (delta 0), reused 0 (delta 0), pack-reused 0', 'To https://github.com/park5430/git_practice2.git', '* [new branch] main -> main', and 'branch \'main\' set up to track \'origin/main\'.'. The terminal window has a standard Windows title bar and a file explorer sidebar on the left.

내 깃헙 계정에서 저장코드와 저장 히스토리 전부 받는걸 클론이라 합니다.

12. 클론할 때 원하는 폴더를 들어가서 git bash 실행한 뒤에

git clone https://github.com/park5430/git_practice2.git 입력

git clone https://github.com/park5430/git_practice2.git .

git clone에서 뒤에 . 붙으면 생기는 차이:

- . 없이 클론하면 저장소 이름까지 전부 클론되고
- . 찍으면 저장소 이름 없이 바로 파일이 저장됩니다.

13. 클론 받아와서 파일을 계속 작업 합니다.

역시 git add 파일이름

git commit -m “파일설명” 으로 지속적으로 버전 올리며 작업해요

14 그다음에 `git push origin main` 으로 작업한 파일을 올리면
최초작업 모음폴더 (v3) --> 내 github계정 업로드 (v3) --> 클론으로
다른폴더에 그동안 작업 받기 (v3)--> 클론 받은데서 추가 작업 반영
(v6) --> `git push origin main` 으로 업로드 (v6)

그러면 아까 최초작업폴더의 (v3)까지의 버전은 어떻게 최신화???
일일이 클론???? --> `pull` 명령어로 최신 작업 분을 적용합니다. + 히
스토리도 저장

15. 최초 작업폴더 가서 `git bash` 켜고
`git pull origin main` 하면 내가 작업한 최신작업분 업데이트