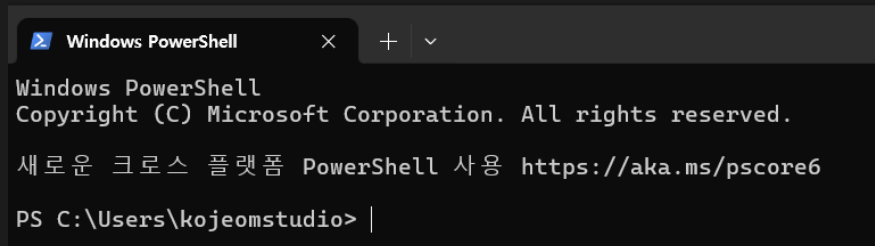

작업 흐름

1. 로컬에 git 리포지터리 생성
2. 초기화 및 세팅
3. Github에 리포지터리 생성
4. 로컬 리포지터리에서 origin을 github 리포지터리로 설정
5. 로컬에서 origin으로 push
6. 연동 확인

로컬 repo 생성 후 github 연동 과정

1. `mkdir C:\workspace\test_repo` # 작업 디렉터리 생성
2. `cd C:\workspace\test_repo` # 작업 디렉터리로 이동
3. `git init` # Git 리포지터리 초기화
4. `echo "Hello GitHub!" > README.md` # 샘플 파일 생성
5. `git add .` # 변경된 파일 스테이징
6. `git commit -m "Initial commit"` # 커밋
7. `git remote add origin git@github.com:kojeomstudio/test_repo.git` # 원격 리포지터리 추가 (SSH 방식)
8. `git branch -M main` # 현재 브랜치를 main으로 변경
9. `git push -u origin main` # 로컬 main 브랜치를 원격 main 브랜치로 푸시



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Users\kojeomstudio> |
```

p.s. windows 10 / power shell 에서 진행.

Github 리포지터리 생성

Type ▾

Language ▾

Sort ▾

New


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner *


Repository name *

 kojeonstudio ▾


/

Great repository names are short and memorable. Need inspiration? How about [effective-octo-enigma](#) ?

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

Create repository

- public 또는 private 으로 설정.
- .gitingore 및 license 설정 시 초기 파일이 자동으로 해당 리포지터리에 생성.

3

로컬과 원격간의 이력(history)에 간격이 있는 경우.

```
PS C:\workspace\test_repo> git push -u origin main
To github.com:kojeomstudio/test_repo.git
! [rejected]        main -> main (fetch first)
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

`git pull origin main --allow-unrelated-histories`

(서로 다른 이력을 강제로 병합 / 원격 main 브랜치를 로컬의 main 브랜치와 병합 시도)

충돌점이 있는 부분을 확인 후 병합 -> 이후 push를 진행.