

▶ 나의 [py.] 파일을 내 GitHub 에 업로드하기

로컬 시스템에서 Git 저장소를 초기화하고, 파일을 추가한 다음, 원격 저장소에 커밋을 푸시하는 단계

1. `git init` : 현재 디렉토리를 Git 저장소로 초기화

2. `git add 파일명` : 변경사항이 있는 특정 파일을 스테이징 영역에 추가

ex) `git add 230914-넣기연습.py`

3. `git commit -m "내용"` : 스테이징 영역에 있는 변경사항들을 커밋

ex) `git commit -m "처음 넣는 파일"`

4. `git remote add origin 레파지토리의 주소` : 원격 저장소를 추가

ex) `git remote add origin https://github.com/username/abc.git` (예시)

5. `git push -u origin master` : 커밋된 변경사항을 원격 저장소에 푸시

추가.`git status` = 현재 파일의 상태를 보는 것

한번 연결된 [GIT]의 과정은 간략해 진다.

1 단계 - `git init`

2 단계 - `git add 파일명`

3 단계 - `git commit -m "내용"`

4 단계 - `git push`

(n/n)완료 메세지 떴야 함

안될 시, `git push` 해보고 `git push set-upstream origin master` 먼저 수행

※새로운 저장소를 만들 경우, `git push -u origin master` 대신 `git push -u origin main` 을 사용

※브랜치의 이름을 확인하려면 `git branch` 명령어를 사용

▶ GitHub 에서 내 컴퓨터로 가져오기

`git pull` : 원격 저장소의 최신 변경사항을 로컬 저장소로 가져온다.

다른 사람이 수행한 변경사항을 로컬에 반영할 때에도 사용

▶ Git Branch 이동 및 협업

- `git checkout <브랜치 이름>` : 다른 브랜치로 이동하고 싶을 때 사용

- `git checkout -b <새 브랜치 이름>` : 새로운 브랜치를 생성하고 해당 브랜치로 이동

▶ 공동 작업 방법

- 프로젝트에 참여하기 위해 `git clone <레포지토리 주소>`를 사용하여 전체 프로젝트를 로컬 시스템에 복제.

- 작업을 시작하기 전에 `git pull` 로 최신 변경사항을 가져온다.

Pull Request : 협업 시, 직접 `main` 이나 `master` 에 푸시하는 대신 'Pull Request' (PR)를 사용하는 것이 일반적인 프로세스

PR 을 통해 다른 팀원들이 코드 리뷰를 수행하고, 통합하기 전에 변경사항을 논의 가능

※코드 리뷰와 피드백 : 코드 리뷰 과정은 협업의 핵심 부분

※GitHub 의 PR 시스템을 통해, 팀원들은 코드에 대한 피드백을 제공하고, 개선점을 논의할 수 있다.

- 작업 후, 변경사항을 `git add` , `git commit` , `git push` 순으로 적용

- 팀원의 최신 변경사항을 받아오려면 `git pull` 을 사용