

# Git-Group Project

## 협업하기





## GitHub

## ➤ GitHub 공동 작업자 추가하기

- 해당 repository로 들어가서 Settings - Access - collaborator – Addpeople로 추가  
**(반드시 초청 받은 팀원들은 각자 승인을 눌러 사용한다)**

The screenshot shows the GitHub repository settings page for 'ohseung-web / netflixPjt2004'. The 'Collaborators' section is highlighted with a red box, and the 'Settings' button in the top navigation bar is also highlighted with a red box. In the 'Who has access' section, it shows a public repository with 2 collaborators. The 'Manage access' section at the bottom right has a green 'Add people' button highlighted with a red box.

ohseung-web / netflixPjt2004

Type ⏎ to search

Code Issues 2 Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

PUBLIC REPOSITORY

This repository is public and visible to anyone.

Manage

DIRECT ACCESS

2 have access to this repository. [1 collaborator](#), [1 invitation](#).

Manage access

Add people

Select all

Type ▾

Find a collaborator...



## ➤ 조장이 할 일

- ① 조원이 push할 수 있게 초대하기
- ② 저장소 페이지 → Settings 탭 클릭
- ③ 왼쪽 메뉴에서 Collaborators 선택  
(주소는 <https://github.com/조장이름/레포/settings/access>)
- ④ "Manage access" 화면에서 → Add people 버튼 클릭
- ⑤ 조원의 GitHub 아이디(username) 입력
- ⑥ "Add collaborator" 클릭
- ⑦ 조원은 이메일 또는 GitHub 알림으로 초대 수락해야 한다.



➤ 조원이 해야 할 일

- ① 조장이 보낸 초대 수락
- ② GitHub 로그인
- ③ 알림(Notification) 또는 이메일에서  
"You've been invited to collaborate on..." 클릭
- ④ Accept invitation 클릭 → 이제 push 권한이 생긴다



➤ 조장이 초대 완료 후, 조원들이 작업하는 기본 흐름

① 최신 코드 받기

git pull origin main

② 내 부분 개발 (예: 로그인 기능)

# 코드 수정 후

③ 변경사항 등록

git add .

④ 커밋 메시지 작성

git commit -m "A조원 : 로그인 기능 추가"

⑤ 조장 저장소에 업로드

git push origin main



## ➤ Push 하는 과정에서 아래 오류가 출력될 때 작성

- 조원 A가 App.jsx를 수정하고 아직 push 안 한 상태에서 조원 B가 같은 파일을 수정해 push하면 A가 git push할 때 오류가 발생한다.

### ➤ 오류 출력

```
! [rejected] main -> main (fetch first)
error: failed to push some refs to ...
hint: Updates were rejected because the remote contains work that you do
not have locally.
```

## ➤ 오류 출력 될 때 해결 방법 아래 명령어를 입력한다.

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

또는

```
git pull origin main --rebase
```



- 충돌 해결 후 다시 아래와 같이 push한다.

```
git add .
```

```
git commit -m "충돌 해결"
```

```
git push origin main
```