

Git-Group Project 협업하기





GitHub

➤ GitHub 공동 작업자 추가하기

- 해당 repository로 들어가서 Settings - Access - collaborator - Addpeople로 추가

ohseung-web / netflexPjt2004

Code Issues 2 Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Who has access

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

☐ Select all

Type

Find a collaborator...

Add people



GitHub

➤ GitHub 공동 작업 버전 관리 하기

- ① 협업을 할 때에는 각기 다른 기능을 작업하는 경우가 대부분이다.
- ② 이럴 때는 main 브랜치에 있는 문서들과 섞이지 않도록 새브랜치를 만들어 버전관리를 하며 협업한다.
- ③ 각 팀원의 새로운 브랜치에서도 원격저장소에 push할수 있기때문이다.
- ④ 원격 저장소에 다른 팀원들의 commit이 추가되어 있는지 확인하기 위해 git pull 명령을 사용하여 최신 수정 내역을 당겨온다.

```
$git pull
```

- ⑤ 다른 기능 및 작업을 위해서 지역저장소에 새로운 브랜치를 만들고 새로운 브랜치로 이동(**checkout**)한다.

```
$git branch -b team1
```

이 과정에서 브랜치명에 대한 브랜치가 이미 있다면 해당 **브랜치로 이동(checkout)**한다.



GitHub

➤ GitHub 공동 작업 버전 관리 하기

- ⑥ 브랜치를 이동했다면 문서를 만든 후 commit 한다.

```
$vim viewtest03.txt  
$git add viewtest03.txt  
$git commit -m "viewtest03"
```

- ⑦ 원본 repo 수정 사항을 내 컴퓨터에 반영

```
$git pull upstream(update) team1
```

- ⑧ 원격 저장소에 브랜치까지 함께 push 한다. git push 명령 뒤에 origin 브랜치 를 적어 주면 원격저장소(origin)에 새로운 브랜치를 push한다는 의미이다.

```
$git push origin team1
```



GitHub

➤ GitHub 공동 작업 버전 관리 하기

⑨ Main으로 merge(병합) 할 때 conflict(충돌)때문에 생기는 오류를 해결

❖ Conflict란? Github의 다른 branch 에서 같은 파일의 같은 줄에 작업을 할 때 코드가 중복되어 발생하는 충돌 상태를 의미한다. Git으로 프로젝트를 관리하게 되면 자주 발생하는 부분이다.

```
$git switch [merge 하려는 브랜치]
$git pull origin [merge 하려는 브랜치]
$git switch [conflict(충돌) 브랜치]
$git merge [merge 하려는 브랜치]
```

```
$git add .
$git commit -m 'comment'
$git push origin [conflict 브랜치]
```



GitHub

➤ GitHub 계정 세팅

- 공용 PC 경우 이미 세팅된 다른 사람의 git 정보가 있을 시 제거하는 방법.

```
git config --global user.email "이메일주소"  
git config --global user.name "본인이름"
```

- Git checkout, update, fetch, merge, pull 개념이해

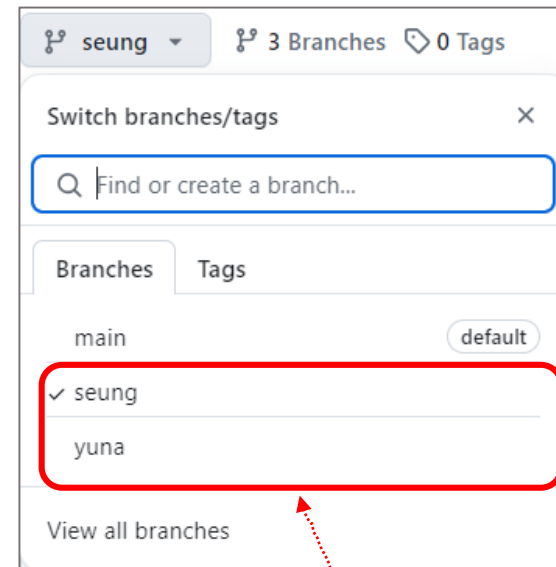
➤ 자주 이용하는 branch 명과 뜻 보기

- ✓ 리더 : Git Branch 규칙과 순서

리더가 큰 틀을 만들고 공통 파일을 업로드하여 팀원이 받을 수 있도록 진행한다.

- ✓ 저장소 등록하고 Push 진행하기

→ 이후 본인의 브랜치를 checkout 명령어로 별도로 만들어서 작업 수행하기





GitHub

➤ 팀원 : Git Branch 규칙과 순서

- Git 작업 터미널 명령어
- GitHub UI 활용 방법 : 작업 수행 후 push merge 진행하기
- 터미널 활용 방법 : 작업 수행 후 push merge 진행하기

➤ 터미널 merge 에러 해결방법

- merge 에러 발생 시 `-allow...` 를 추가하여 병합 오류를 해결할 수 있다.
- 같은 파일의 같은 부분을 병합하려 할 경우 아래와 같은 에러 발생할 가능성 높음

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