



멋쟁이사자처럼 1차시

OT 및 Github 연동

커리큘럼

1차시 (03.13) - OT 및 Github 연동

2차시 (03.20) - 프론트엔드 기초[1]

3차시 (03.27) - 프론트엔드 기초[2]

4차시 (04.03) - CSS 심화

5차시 (05.01) - Javascript 기본

6차시 (05.08) - Javascript 심화

7차시 (05.15) - React [1]

8차시 (05.22) - React [2]

9차시 (05.29) - React [3]

1차시 목차

멋사 프론트엔드 정의 및 설명



학습 목표



공부 방식 소개



짝 배정



Github 연동



교류의 시간



‘멋쟁이사자처럼’은 뭘까..?



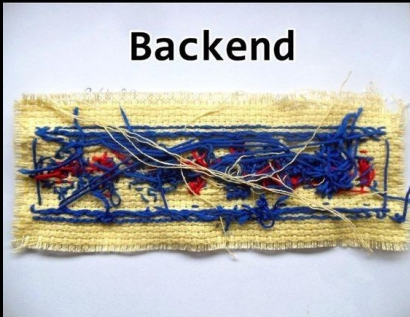
OT에서 했으니 간단하게

‘프론트엔드’는 뭘까?

Frontend



Backend



Front-end

Back-end

‘프론트엔드’는 뭘까?



프론트엔드 개발자

‘프론트엔드’는 뭘까?



Front-End
클라이언트

서비스 요청

요청에 응답



기본 쿠
고맙다쿠
화났다쿠
밥먹자쿠
...

Back-End
서버 및 데이터베이스

‘프론트엔드’는 뭘까?



Front-End

=> 웹사이트에서 사용자들이 사용하는 화면과 같은 앞단.

=> 데이터를 요청하는 클라이언트 영역.



기본 쿠
고맙다쿠
화났다쿠
밥먹자쿠
...

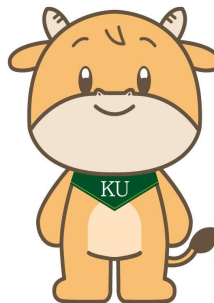
Back-End

=> 눈에 보이지 않는, API와 같은 화면 뒷단.

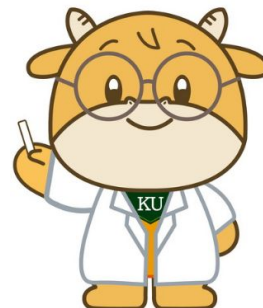
=> 데이터 요청에 응답하는 서버 영역.



공부가 즐거운 쿠



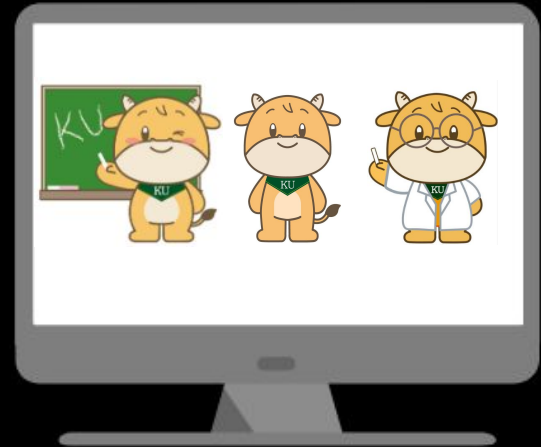
쿠



레지던트 쿠



Back-End



Front-End



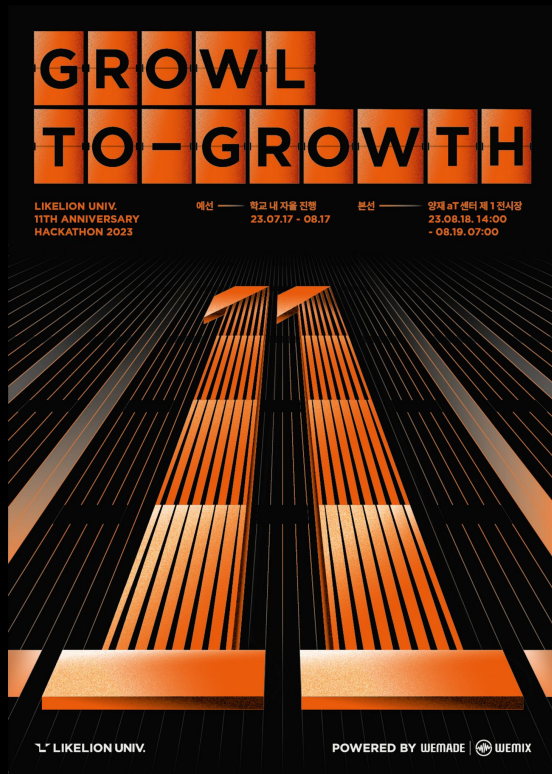
Front-End
클라이언트



Back-End
서버 및 데이터베이스

기본 쿠
고맙다쿠
화났다쿠
밥먹자쿠
...

학습 목표



12기 아이디어톤

★ 주제

💡 사회적, 보편적 문제를 해결하는 서비스

여러분의 주변에서 발생하는 다양한 사회적, 보편적인 문제는 어떤 것이 있을까요?

그러한 문제를 해결하거나 예방할 수 있는 서비스를 위해 상상력을 펼쳐주세요!

👥 팀 구성

💡 최소 3명 ~ 6명 구성

소속 학교 내에서 자신의 뜻을 함께할 팀원을 구해보세요.

팀원이 소집되면 본선 수상까지 팀을 리드할 팀장을 선발해주세요.

11기 아이디어톤 주제

내용	일정	비고
팀 구성	4/29(월) ~ 5/5(일)	- 각 학교 내 팀 구성
주제 공개	5/6(월) 14시	- 홈페이지+카톡방 공지
1차 예선 진행 및 심사	5/6(월) ~ 5/20(월)	- 해당 일정동안 각 학교 내에서 진행 - 심사기간 자율 설정 - 단, 제출 마감 기한 엄수
작품 제출 마감 및 2차 예선 진출팀 최종 확정	5/21(화) 14시	- 전체 참가 팀 작품 제출(학교별 드라이브)
2차 예선 진행 및 심사	5/22(수) ~ 5/24(금)	- 해당 일정 내에 각 그룹별 심사 진행
3차 본선 진출팀 확정	5/28(화) 14시	- 각 그룹별 1개팀 선정
3차 본선 진행	6/1(토) 14시~18시	- 오프라인 진행(멧사 광화문 오피스) - 발표자 필수 참여



일정 : (본선) 2024.06.01(토), 상세내용 좌측 표 참고

주제 : 추후 공개

팀 빌딩 인원 : 3~6명 (*기준과 동일)

12기 해커톤

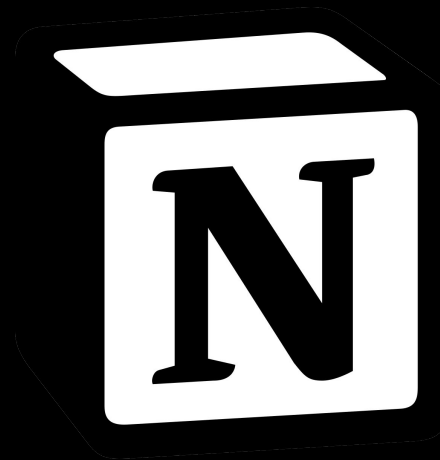
멋쟁이사자처럼 대학의 꽃, 해커톤!
해커톤에서 열정을 불태워 보고, 좋은 프로젝트가 나온다면
완성도를 높여 데모데이까지 도전해 보세요!

내용	일정	비고
주제 공개	7/12(금) 14시	- 홈페이지+카톡방 공지
팀 빌딩 및 개발	7/12(금)~8/12(월)	- 각자 팀을 꾸려 해커톤 주제에 맞는 프로덕트 개발
오프라인 해커톤 본선	8/13(화)~8/14(수), 무박 2일 간 진행	- 오프라인 진행(양재 aT센터) - 해커톤 참여 여부 → 수료 기준 반영



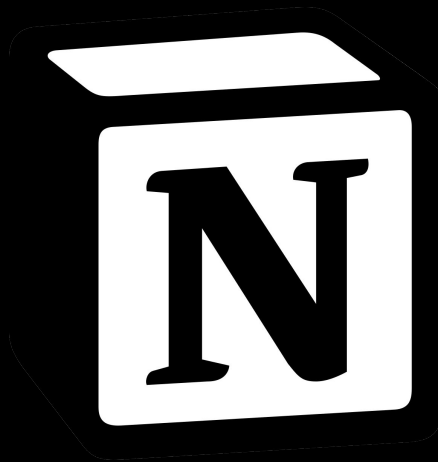
- 일정 : 2024.08.13(화)~08.14(수), 무박 2일
- 주제 : 추후 공개
- 팀 빌딩 인원 : 최대 6명 (*기존과 동일)

공부 방식



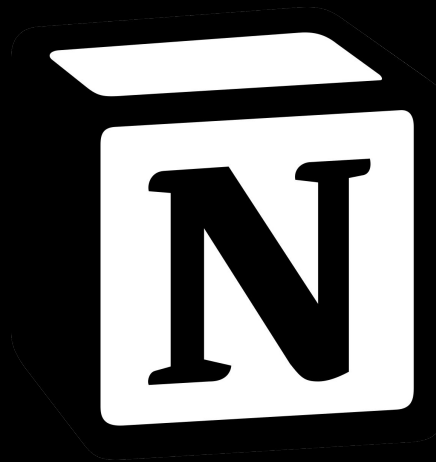
토스뱅크 1000-4516-9790

공부 방식 - 따라해봅시다!



복습 대시보드에 글 올려보기

과제 1: 2주차 대비 과제



~3/17 일요일 오후 9시
노션 템플릿 1-1. ~ 1-5.

과제 1 짝 배정

구유진	이다연
김나영	천희영
김현수	이현주
김지현	신가은

Github 연동

‘Git’ & ‘Github’



‘로컬’에서 관리되는
버전관리 도구 및 시스템



‘클라우드’를 통해 관리되는
버전 관리 시스템

‘Git’ PUSH



로컬에서 작성한 코드

코드 저장소

'Github' PULL



git



pull



남이 작성한 코드
수정

코드 저장소에 있는
남이 작성한 코드

‘Git’ & ‘Github’



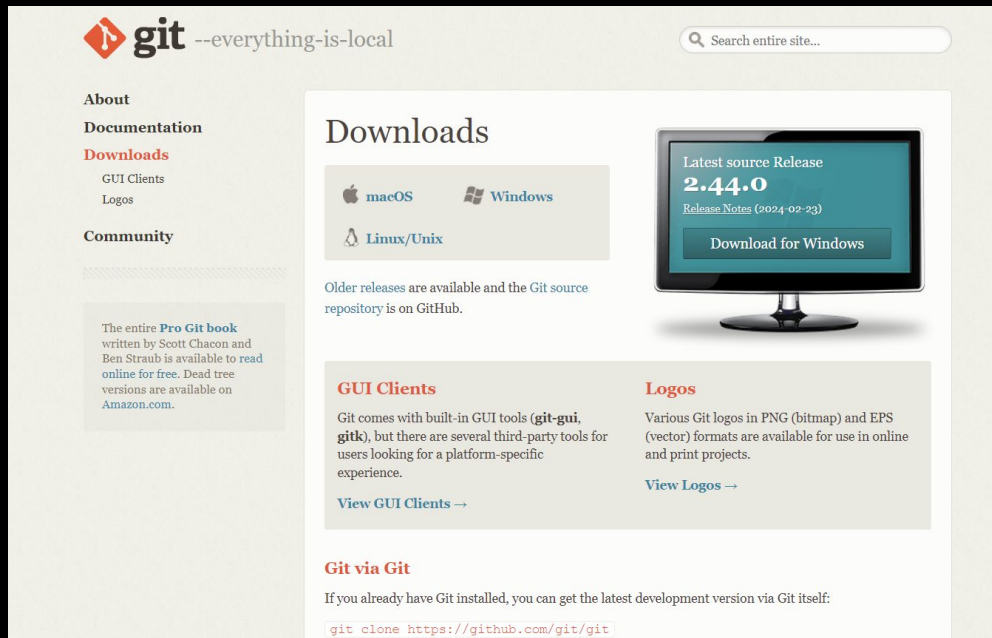
‘로컬’에서 관리되는
버전관리 도구 및 시스템



‘클라우드’를 통해 관리되는
버전 관리 시스템

‘Git’ 설치하기

<https://git-scm.com/downloads>



The screenshot shows the Git website's 'Downloads' page. The header includes the Git logo and the tagline '--everything-is-local'. A search bar is located at the top right. The left sidebar contains links for 'About', 'Documentation', 'Downloads' (highlighted), 'GUI Clients', 'Logos', and 'Community'. The main content area features a 'Downloads' section with links for macOS, Windows, and Linux/Unix. A central image of a monitor displays 'Latest source Release 2.44.0' and a 'Download for Windows' button. Below this, there are sections for 'GUI Clients' and 'Logos', each with descriptive text and a 'View' link. At the bottom, the 'Git via Git' section provides a command to clone the repository: `git clone https://github.com/git/git`.

git --everything-is-local

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

Downloads

macOS Windows Linux/Unix

Latest source Release
2.44.0
Release Notes (2024-02-23)
Download for Windows

Older releases are available and the Git source repository is on GitHub.

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.
[View GUI Clients →](#)

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.
[View Logos →](#)

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```



The screenshot shows the 'Git 2.40.0 Setup' window. The title bar includes the Git logo and the text 'Git 2.40.0 Setup'. The window has standard Windows window controls (minimize, maximize, close). The main content area is titled 'Information' and contains the text 'Please read the following important information before continuing.' Below this, a message states 'When you are ready to continue with Setup, click Next.' A large text box displays the 'GNU General Public License' (Version 2, June 1991), including the copyright notice and the preamble. At the bottom right, there are 'Next' and 'Cancel' buttons. A mouse cursor is pointing at the 'Next' button.

Git 2.40.0 Setup

Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

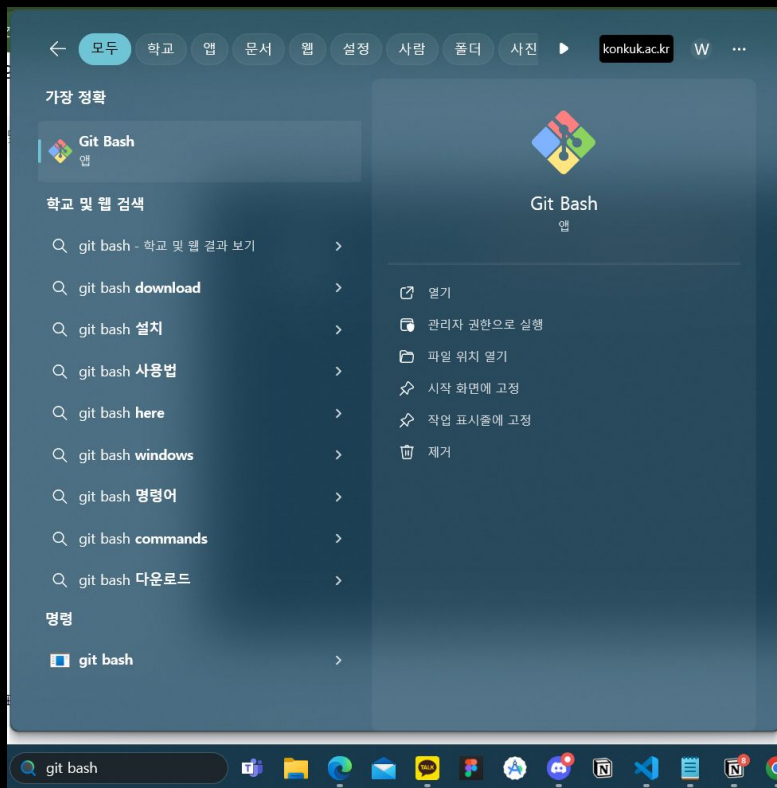
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

<https://gitforwindows.org/>

Next Cancel

'Git Bash' 실행 및 최초설정



최초설정

```
USER@LAPTOP-MN3TJCC1 MINGW64 ~ (231220_JH)
$ cd LIKELION12

USER@LAPTOP-MN3TJCC1 MINGW64 ~/LIKELION12 (231220_JH)
$ git config --global user.name "cheerjihyeon"

USER@LAPTOP-MN3TJCC1 MINGW64 ~/LIKELION12 (231220_JH)
$ git config --global user.email gimjihyeon111@gmail.com
```


설정확인

```
USER@LAPTOP-MN3TJCC1 MINGW64 ~/LIKELION12 (231220_JH)
$ git config --list
```

```
user.name=cheerjihyeon
user.email=gimjihyeon111@gmail.com
```

‘Github’ 회원가입

<https://github.com/login>



Sign in to GitHub

Username or email address

Password

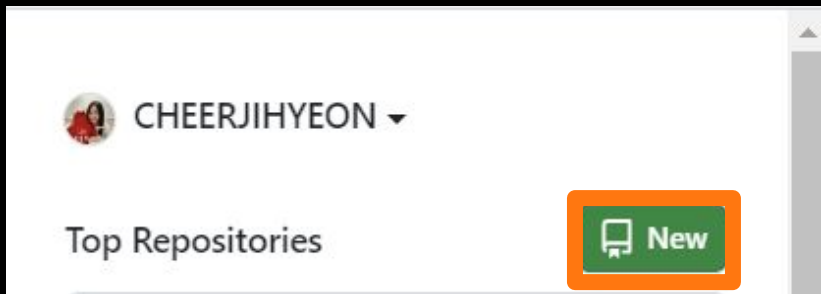
[Forgot password?](#)

Sign in

[Sign in with a passkey](#)

New to GitHub? [Create an account](#)

‘Github’ 레포지토리 생성




1. 로그인 후 홈 화면에서
왼쪽 상단에 있는
초록색 New 버튼 클릭


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 *


 CHEERJIHYEON / LIKELION12_FE

 LIKELION12_FE is available.

Great repository names are short and memorable. Need inspiration? How about didactic-funicular ?

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

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

Choose a license

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.

2. Repository name을 작성 후 저장소 생성

로컬 저장소와 원격 저장소 연결

1. **\$ cd [폴더이름]**
git bash 실행한 후
로컬 저장소에 접속
(remote에 업로드할 폴더로 이동)
cd : change directory 즉 디렉토리
이동 명령어
2. **\$ git remote add origin**
[자신의깃허브경로]
원격 저장소 등록
\$ git remote -v : 원격 저장소가 잘
등록되었는지 확인
3. 로컬에서의 commit
\$ git init (초기 실행 시)
\$ git add .
\$ git commit -m "first commit"

```
USER@LAPTOP-MN3TJCC1 MINGW64 ~ (231220_JH)
```

```
$ cd LIKELION12
```

```
USER@LAPTOP-MN3TJCC1 MINGW64 ~/LIKELION12 (231220_JH)
```

```
$ git remote add origin https://github.com/CHEERJIHYEON/LIKELION12_FE.git
```

```
USER@LAPTOP-MN3TJCC1 MINGW64 ~/LIKELION12 (231220_JH)
```

```
$ git remote -v
```

```
origin https://github.com/CHEERJIHYEON/LIKELION12_FE.git (fetch)
```

```
origin https://github.com/CHEERJIHYEON/LIKELION12_FE.git (push)
```

```
USER@LAPTOP-MN3TJCC1 MINGW64 ~/LIKELION12 (231220_JH)
```

```
$ git add .
```

```
USER@LAPTOP-MN3TJCC1 MINGW64 ~/LIKELION12 (231220_JH)
```

```
$ git commit -m "LIKELION12_FE_firstcommit"
```

로컬 저장소와 원격 저장소 연결

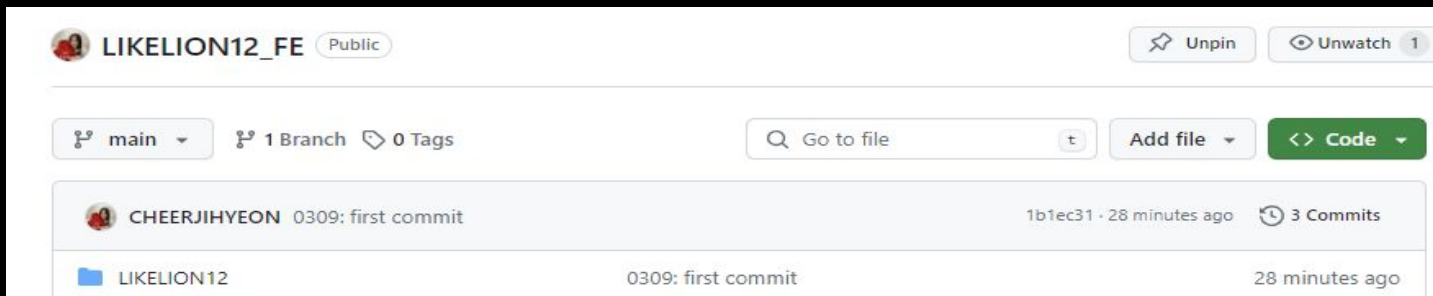
4. git push 실행

\$ git push [원격 저장소의 이름] [push하기 희망하는 로컬저장소의 브랜치 이름]

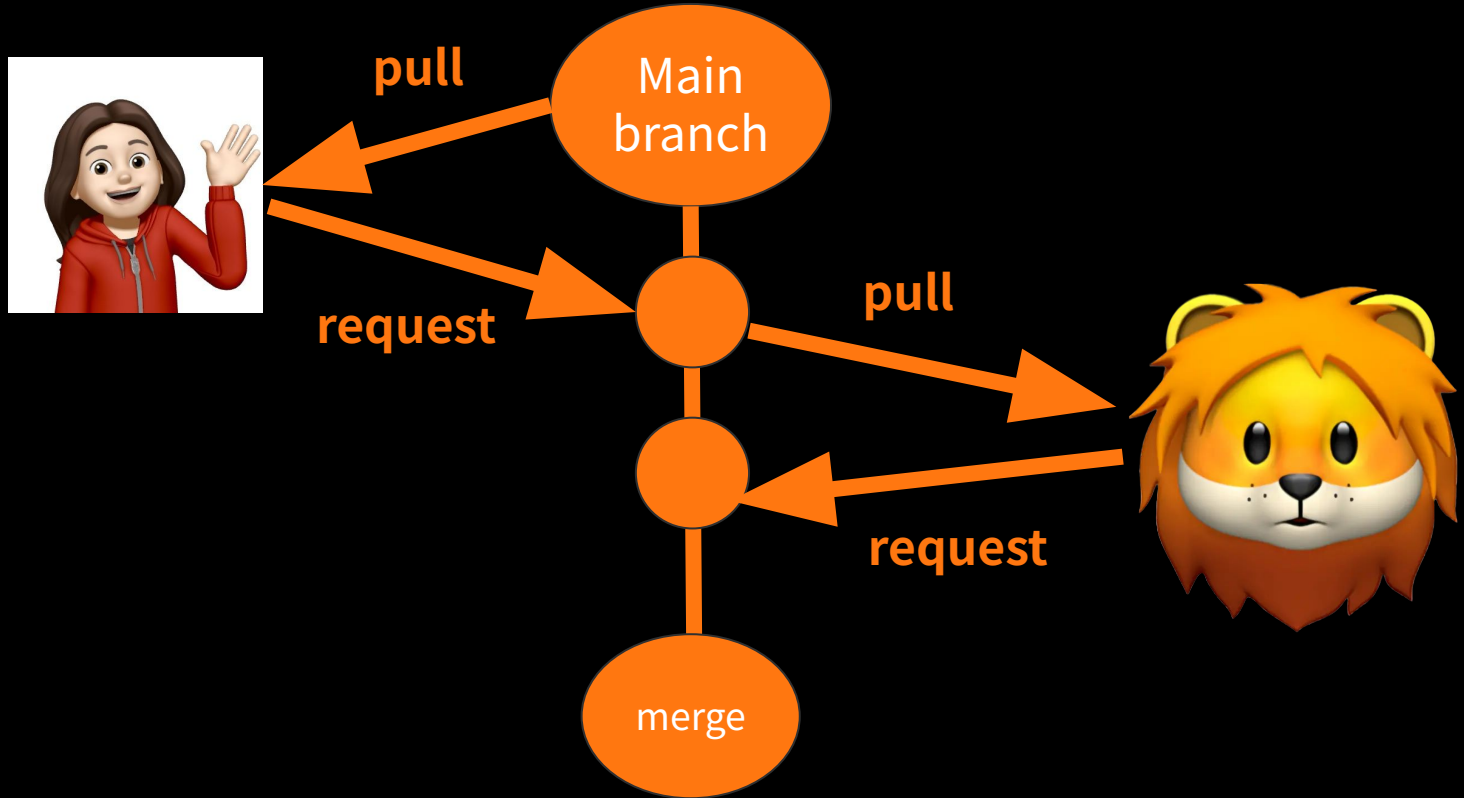
```
USER@LAPTOP-MN3TJCC1 MINGW64 ~/LIKELION12 (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 472 bytes | 118.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/CHEERJIHYEON/LIKELION12_FE.git
1b1ec319..6d0ea042  main -> main
```

로컬 저장소와 원격 저장소 연결

5. github 접속 후 push 된 내역 확인



Pull & Request



LIKELION12_FE Public

Pin Unwatch 1

feature/login had recent pushes 9 minutes ago

Compare & pull request

main 2 Branches 0 Tags

Go to file t Add file Code

Create pull request

Merge pull request

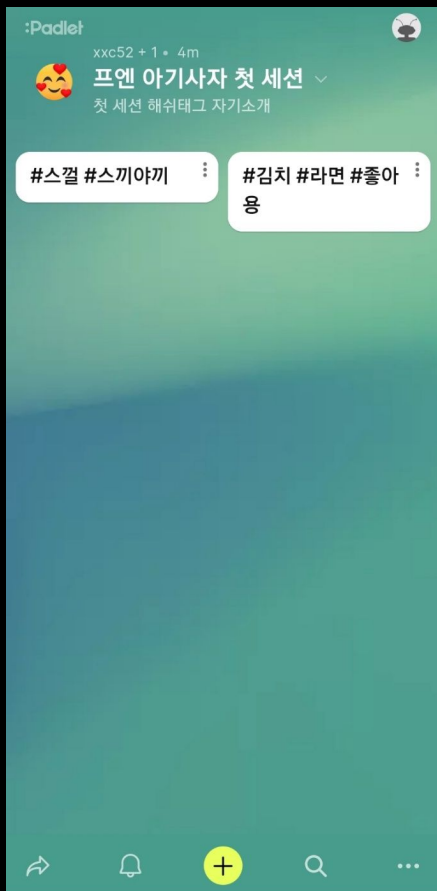
Github 연동 끝

소박한 레크레이션



QR 코드 스캔

https://padlet.com/xxc52/frontend_intro



1. 하단 중앙 + 버튼으로 글 작성
2. 제목에 해쉬태그 2~3개 작성 후 발행
이름, 나이, 전공 작성 금지!

예시

× 📄 발행

☒ 화이트

제목 **#크로스핏 #검정치마 #아오스**

아름다운 내용을 적어보세요...

수고하셨습니다.