

🏃 같이 준비해봅시다!

우리는 깃과 깃허브를 사용해요!



git



🔗 시나그램 깃허브 URL <https://github.com/SinaGram-DSM>

 같이 준비해봅시다!

VCS

Git

Github

 같이 준비해봅시다!

VCS?

 같이 준비해봅시다!

VCS?

버전 관리 시스템
(Version Control System)

🏃 같이 준비해봅시다!

📁 시나그램 발표 폴더



 같이 준비해봅시다!

Git?

 같이 준비해봅시다!

Git?

버전 관리를 할수 있게 도와주는 툴

 같이 준비해봅시다!

왜 해야하나요?

1. 잘못 변경할 시 복구하기 위해
2. 협업할 때 나눠서 작업하기 위해
3. 협업하여 프로젝트를 진행할 때 각자의 코드를 하나로 합치기 위해
4. 코드를 누가, 언제, 어떤 내용을 변경했는지 알기 위해
5. 코드의 변경을 안전하게 진행하기 위해

 같이 준비해봅시다!

Github?

 같이 준비해봅시다!

Github?

git을 사용하는 프로젝트를 지원하는 웹호스팅 서비스

 같이 준비해봅시다!

다섯가지
용어는
꼭 알아두기로 해요

1. 레포지토리(repository) : 저장소
2. 클론(clone) : remote 레포지토리를 로컬로 복사해오는 것
3. 커밋(commit) : 프로젝트의 변경 사항을 레포지토리에 기록하는 것
4. 스테이지(stage) : 커밋하기 직전의 상태
5. 푸시(push) : 커밋한 후 remote 레포지토리로 올리는 것

 같이 준비해봅시다!

레포지토리 (repository) : 저장소

- 1) local : 로컬 컴퓨터의 저장소
- 2) Remote : 서버상의 원격 저장소

 같이 준비해봅시다!

클론 (clone) : remote 레포지토리를 로컬로 복사해오는 것

 같이 준비해봅시다!

커밋 (commit) : 프로젝트의 변경 사항을 레포지토리에 기록하는 것

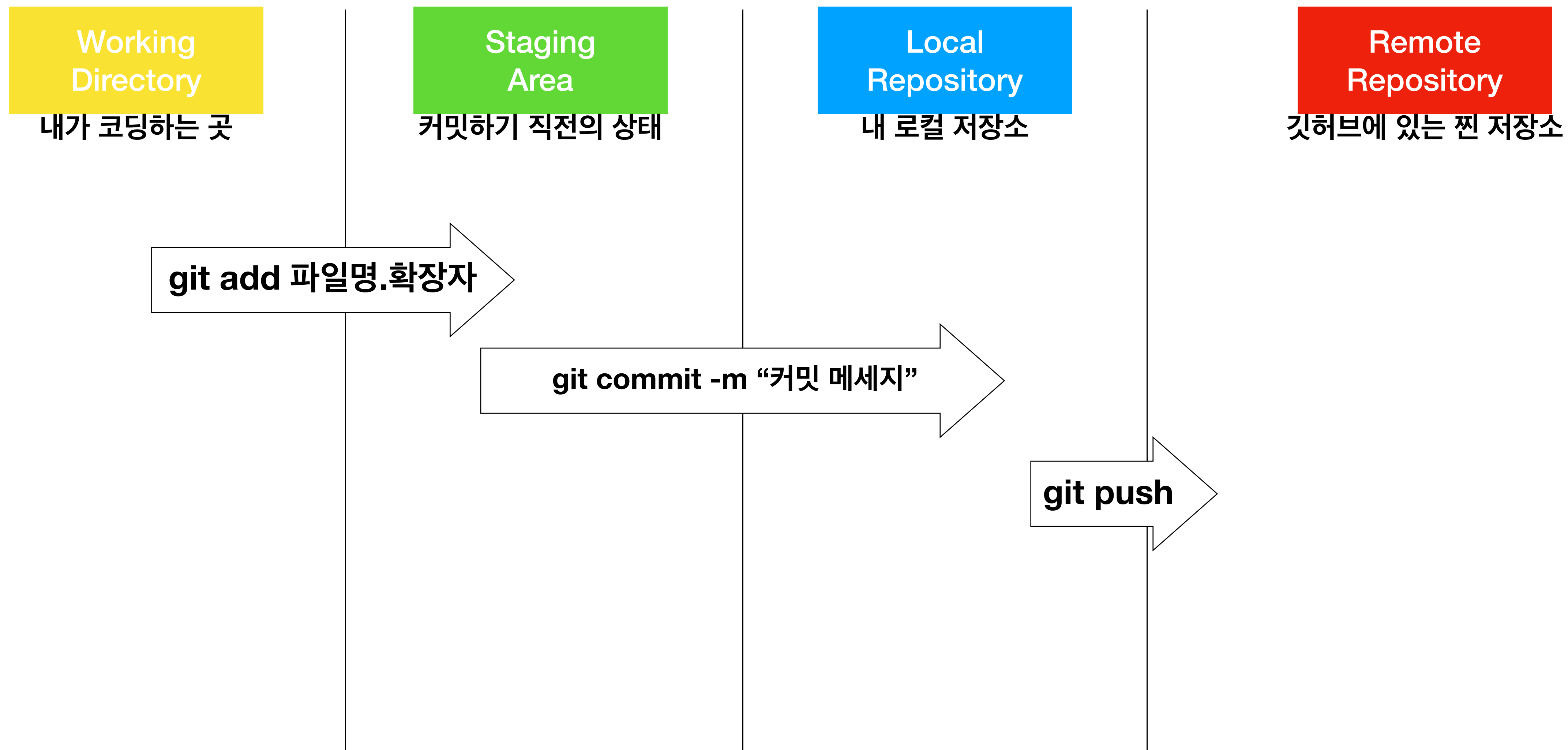
 같이 준비해봅시다!

스테이지 (stage) : 커밋하기 직전의 상태

 같이 준비해봅시다!

푸시 (push) : 커밋한 후 remote 레포지토리로 올리는 것

 같이 준비해봅시다!



🏃 같이 준비해봅시다!

📁 무야호.pdf

💬 커밋메세지 : 그만큼 신나시는 거지

Remote
Repository

깃허브에 있는 찐 저장소

2021년 3월 7일(일)

📁 무하안.pdf

💬 커밋메세지 : ~

2021년 3월 6일(토)

📁 혹시 액션이 어떻게 되는지 아세요?.pdf

💬 커밋메세지 : 노홍철

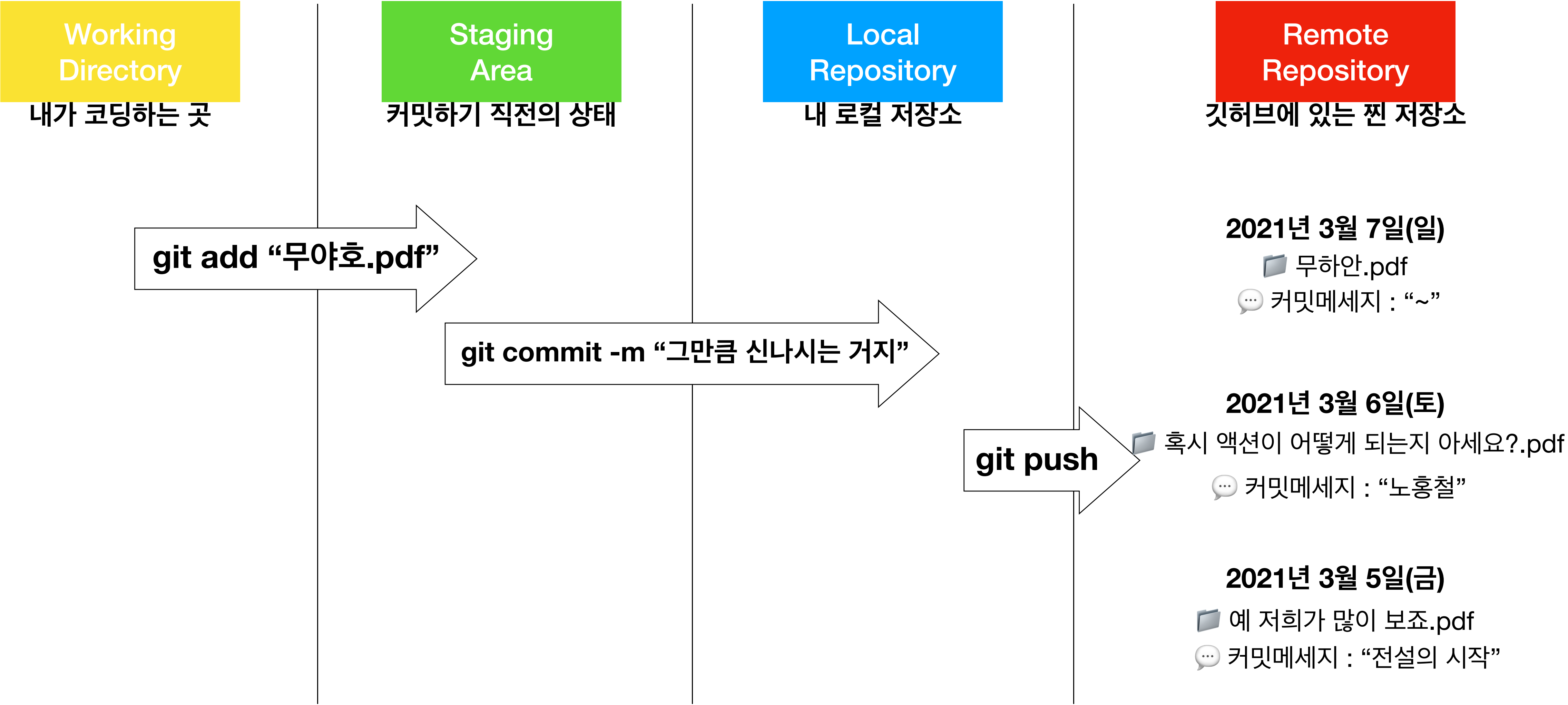
2021년 3월 5일(금)

📁 예 저희가 많이 보죠.pdf

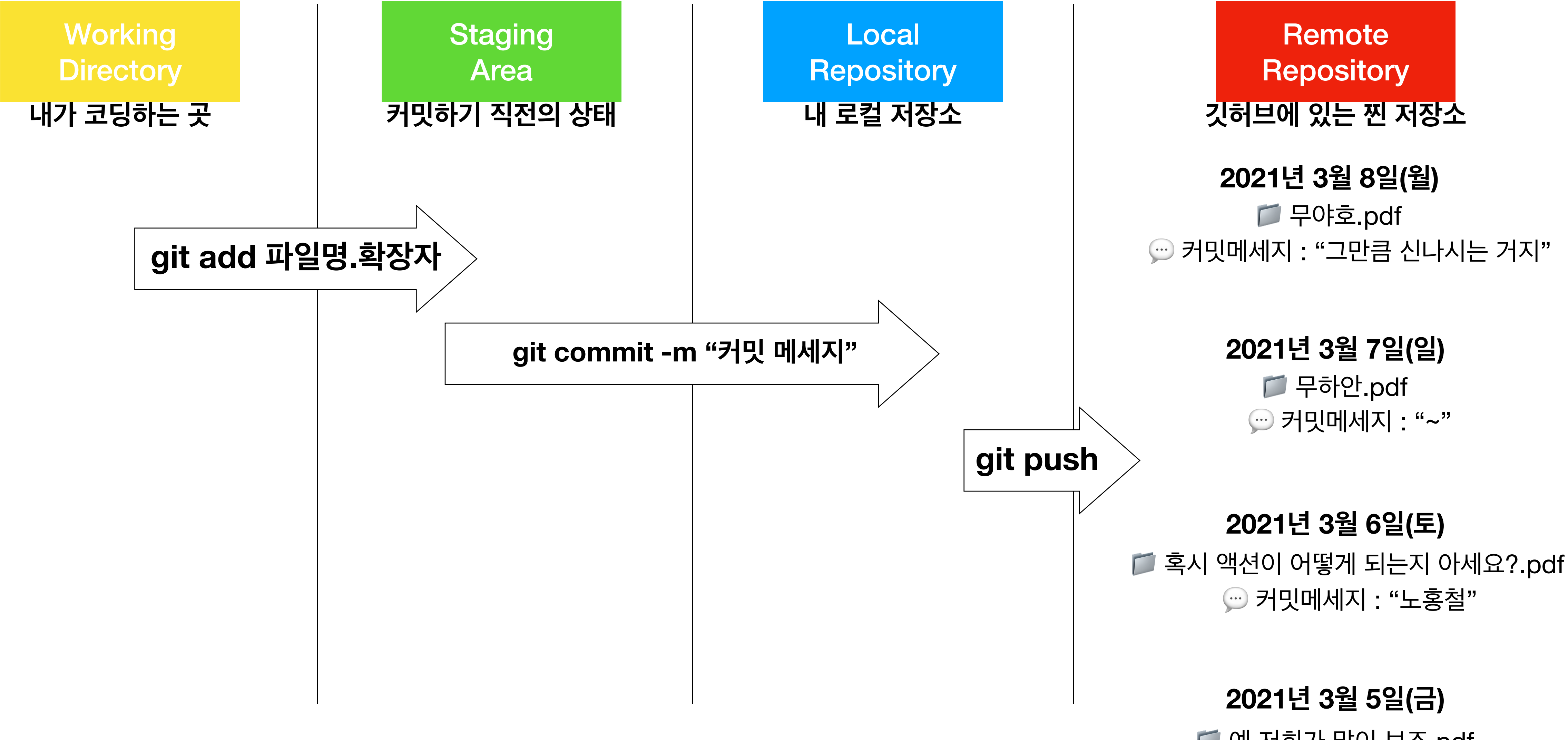
💬 커밋메세지 : 전설의 시작

History

🏃 같이 준비해봅시다!



🏃 같이 준비해봅시다!



 같이 준비해봅시다!

우리도
같이
해봅시다

레포지토리(repository)를 생성하는 방법은 두 가지가 있습니다!

로컬에서
생성하기

깃허브에서 원격 저장소를 만들어
클론하기

 같이 준비해봅시다!

우리도
같이
해봅시다

레포지토리(repository)를 생성하는 방법은 두 가지가 있습니다!

로컬에서
생성하기

깃허브에서 원격 저장소를 만들어
복론하기



🏃 같이 준비해봅시다!

우리도
같이
해봅시다



먼저, 앞으로의 로컬 레포지토리를 모아둘 폴더를 만드는 것을 추천해요!



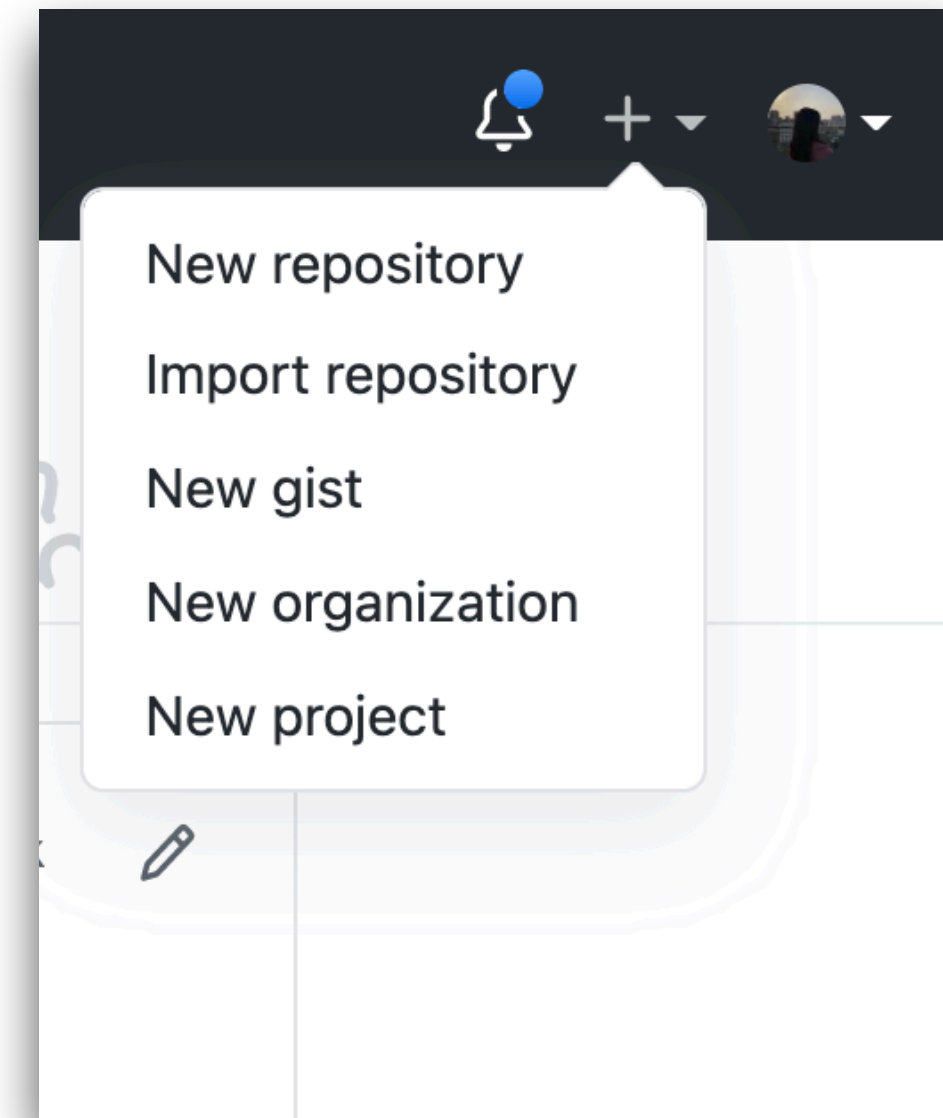
GIT

여러분의 노트북에 GIT 폴더를 하나 만들어주세요

🏃 같이 준비해봅시다!

우리도
같이
해봅시다

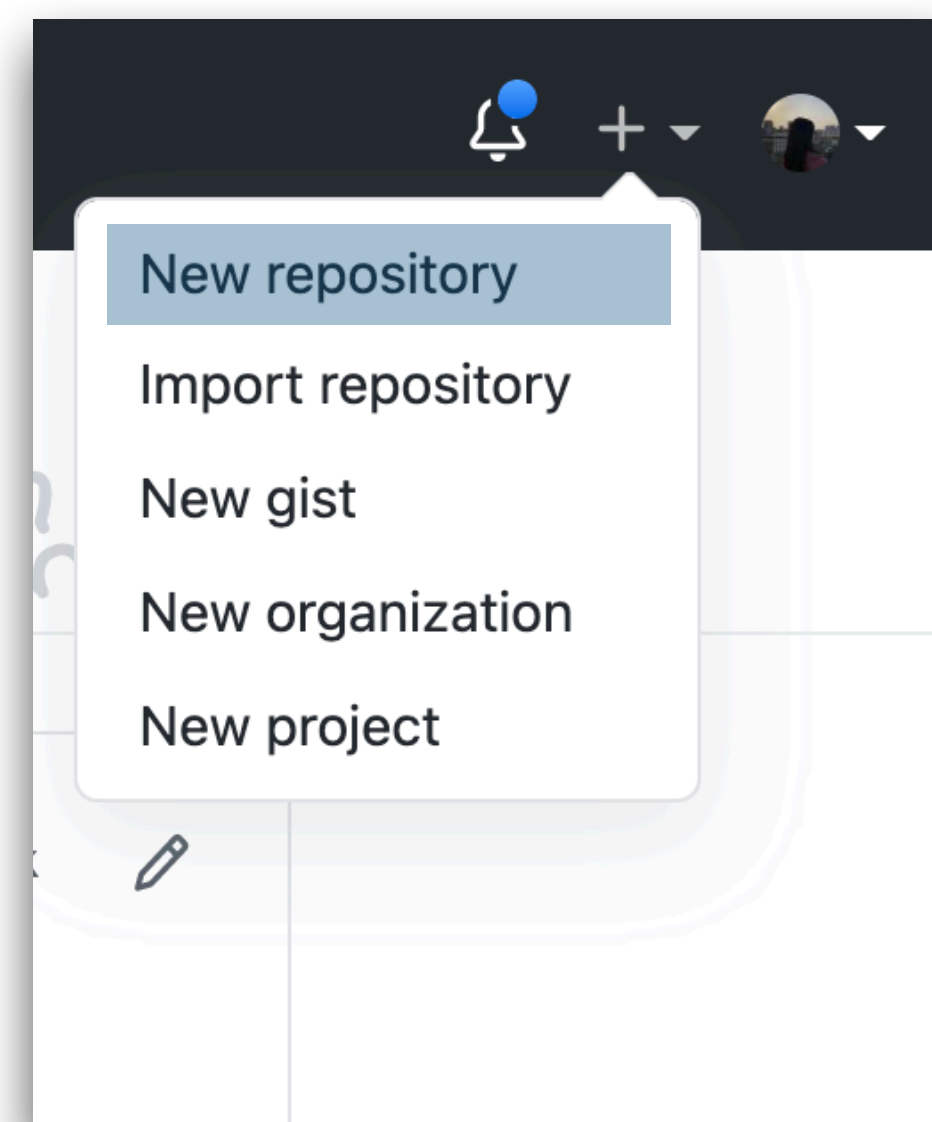
자신의 github에 들어가서 맨 위 오른쪽에 있는 + 버튼을 눌러주세요.



🏃 같이 준비해봅시다!

우리도
같이
해봅시다

New repository를 눌러주세요.



🏃 같이 준비해봅시다!

우리도
같이
해봅시다

Create a new repository

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

Owner *



S-rim ▾

Repository name *

레포지토리_이름 ✓

Great repository name! Your new repository will be created as `-_-`. Suggestion: How about **friendly-disco**?

Description (optional)

레포지토리 설명 적는 곳



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:

Skip this step if you're importing an existing repository.

☒ Add a README file

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

☐ Add .gitignore

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

☐ Choose a license

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

This will set `main` as the default branch. Change the default name in your [settings](#).

Create repository

4. 클릭해주세요!

스크린샷

1. 영어로 작성해주세요

2. public(공개)로 설정해주세요

3. README 파일을 추가해주세요

 같이 준비해봅시다!

 여러분은 github에서 원격 저장소를 생성하는데에 성공했습니다! 
이제 본격적으로 git을 사용해볼까요?

🏃 같이 준비해봅시다!

우리도
같이
해봅시다

git을 사용하는 방법은 두 가지가 있어요

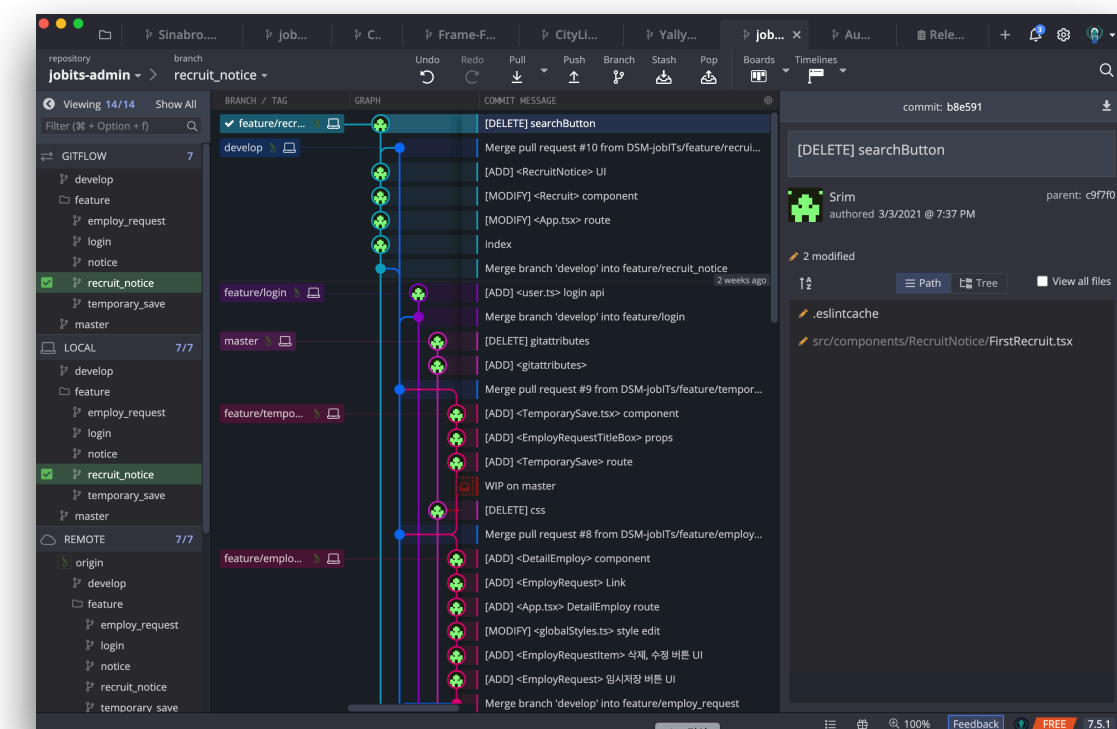
CLI

Command Line Interface



GUI

Graphic User Interface



🏃 같이 준비해봅시다!

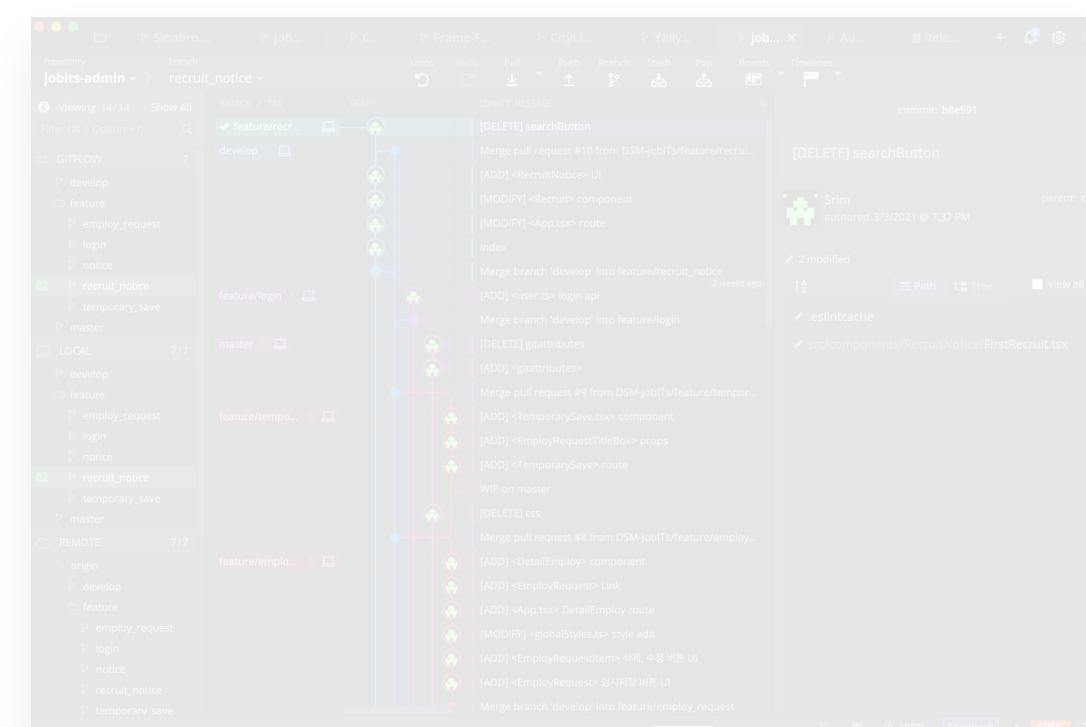
우리도
같이
해봅시다

git을 사용하는 방법은 두 가지가 있어요

CLI
Command Line Interface



GUI
Graphic User Interface



git을 사용할 때 어떤 툴을 사용하는지는 자유!
하지만 이번 오티때는 CLI로 같이 시작해봅시다😊

 같이 준비해봅시다!

우리도
같이
해봅시다

아까 만든 GIT 폴더로 가서 오른쪽 마우스를 클릭해 **git bash**를 켜주세요!

```
git config --global user.name "깃허브 유저 이름"
```

위 명령어를 입력해주세요.

 같이 준비해봅시다!

우리도
같이
해봅시다

아까 만든 GIT 폴더로 가서 오른쪽 마우스를 클릭해 **git bash**를 켜주세요!

```
git config --global user.email "깃허브 이메일 주소"
```

위 명령어를 입력해주세요.

 같이 준비해봅시다!

우리도
같이
해봅시다

확인해볼까요?

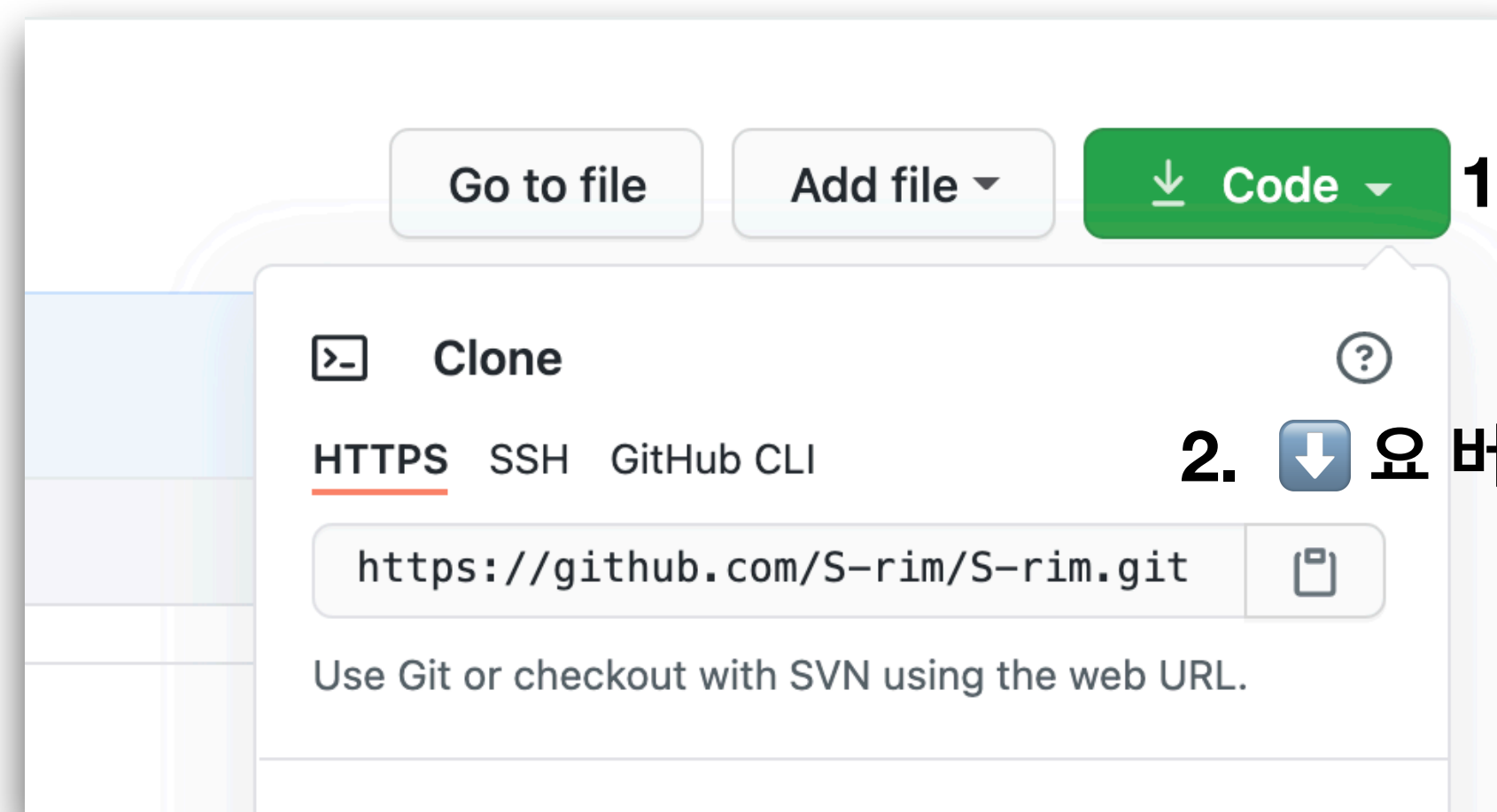
```
git config --list
```

위 명령어를 입력해주세요. 설정한 내용들을 보여줄 겁니다.

🏃 같이 준비해봅시다!

우리도
같이
해봅시다

아까 만들었던 레포지토리에 들어가주세요.



1. Code 버튼 클릭!

2. 📄 요 버튼 클릭!

 같이 준비해봅시다!

우리도
같이
해봅시다

아까 만든 GIT 폴더로 가서 오른쪽 마우스를 클릭해 **git bash**를 켜주세요!

```
git clone 아까 복사한 주소
```

위 명령어를 입력해주세요.

🏃 같이 준비해봅시다!

우리도
같이
해봅시다

GIT 폴더안에 **아까 생성한 레포지토리 이름의 폴더**가 생겼나요?

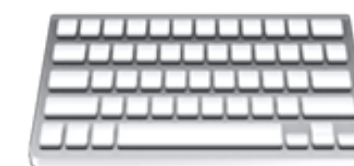


그렇다면 축하합니다! 원격 저장소를 여러분의 로컬로 복사해왔어요 🎉

🏃 같이 준비해봅시다!

우리도
같이
해봅시다

로컬 레포지토리 안에 있는 README.md 파일을 수정해볼까요?



저장하는 것을 잊지 마세요 😊

 같이 준비해봅시다!

우리도
같이
해봅시다

아까 만든 GIT 폴더로 가서 오른쪽 마우스를 클릭해 **git bash**를 켜주세요!

```
cd 여러분이 만든 레포지토리 이름
```

위 명령어를 입력해주세요. **cd**는 폴더에 진입하는 터미널 명령어입니다.

 같이 준비해봅시다!

우리도
같이
해봅시다

이어서 아래의 명령어를 입력해주세요!

```
git add README.md
```

방금 여러분이 변경한 README.md 파일을 **stage** 상태로 변경하는 작업이에요

 같이 준비해봅시다!

우리도
같이
해봅시다

이어서 아래의 명령어를 입력해주세요!

```
git commit -m "여러분이 쓰고 싶은 커밋 메시지"
```

Stage 상태인 변경사항들을 모두 기록하는 작업이에요. 커밋을 할 때는 메시지가 있어야합니다!
오늘은 **Edit README**로 커밋 메시지를 기록해볼까요?

 같이 준비해봅시다!

우리도
같이
해봅시다

이제 거의 다왔습니다!

```
git push
```

로컬에 기록한 커밋을 원격 저장소로 밀어 올려줍니다.

 같이 준비해봅시다!

우리도
같이
해봅시다



깃허브 로그인 창이 뜰 거예요!

깃허브 계정과 비밀번호를 입력해주세요😊

🏃 같이 준비해봅시다!

우리도
같이
해봅시다

로그인에 성공했다면 깃허브로 가서 원격 저장소를 확인해보세요!

| | | |
|---|------------------------|------------------|
|  | S-rim Update README.md | |
|  | README.md | Update README.md |

여러분이 바꾼 내용으로 README.md가 변경되어 있나요?

축하해요! 여러분은 이제 모든 준비가 끝났습니다 🎉

여러분을 도와줄 선배들이 있다는 사실을 잊지마세요!