

파일 시스템 / 터미널 / Git / Vim

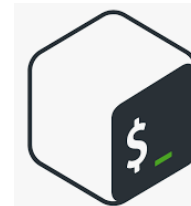
0번째 세션

NEXT X LIKELION 허영범

Detail Curriculum

Terminal

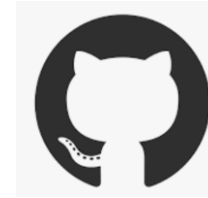
1. 터미널의 기초 명령어 배우기



Detail Curriculum

Git and Github

1. Git 로컬에서 기초 버전관리
2. Github Repository에 푸시하기



| Detail Curriculum

CS Knowledge

CS

1. 파일 디렉토리 구조

| Index

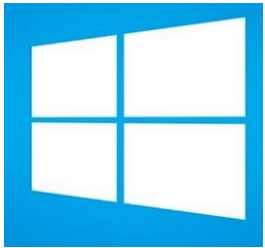
0. 프로그래밍 툴들 설치

1. 파일 시스템 간단하게 알아보기

2. Git 간단한 사용법 알아보기

3. Github Repository 만들고 코드 올려 보기

Git 초기 설정



```
git config --global user.name ${user.name}
```

ex) 자신의 이름이 Kim Sample이면

```
git config --global user.name SampleKim
```

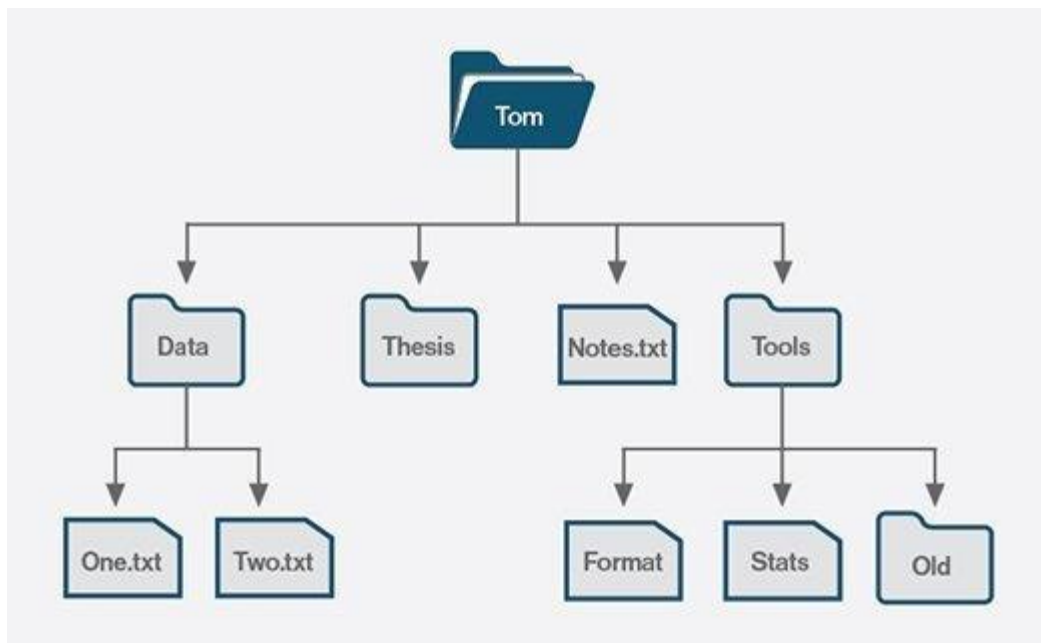
```
git config --global user.email ${user.email}
```

ex) 자신이 깃허브를 가입한 이메일이 sample@korea.ac.kr이면

```
git config --global user.email sample@korea.ac.kr
```

사소한 지식: --global은 전역에 대한 CLI 명령어로 이번 설정이 본인의 컴퓨터 전체 설정에 영향을 줍니다.

File System



디렉토리 구조 : 계층 트리 구조

하위 폴더에 접근하려면 차례대로

Ex) Tom/Tools/Format
Tom/Data/One.txt

CLI (Command Line Interface)



CLI : 텍스트 기반의 명령어 입력으로 시스템 조작
<-> GUI (Graphic User Interface, 여러분들이 사용하는 그것)
자원 소모가 적고, 여러 명령을 커스텀 할 수가 있다.

Windows는 powershell

MacOS / Linux 는 Bash 및 sh 이용

앞으로 개발할 때 매우 필수

CLI (Command Line Interface)

켜는 방법

Windows

Option1. Windows PowerShell 검색



Option2. Git Bash 검색

Mac OS

Spotlight (command + spacebar)

-> Terminal 검색



CLI (Command Line Interface)

Basic Command

ls

cd

mkdir

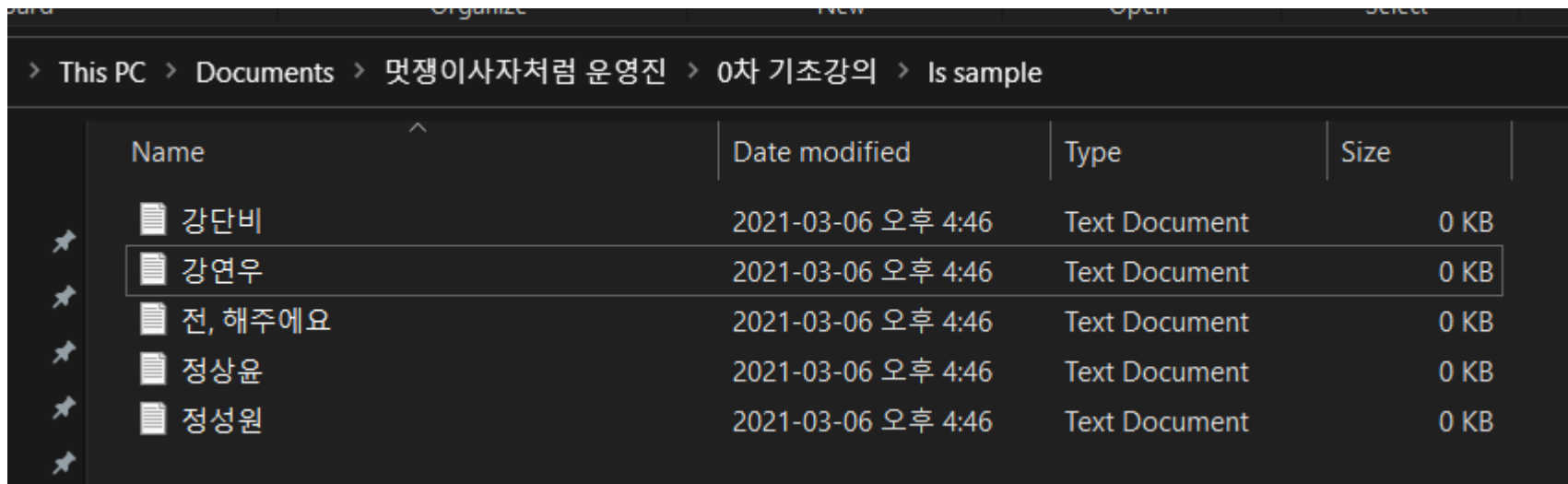
touch

pwd

rm

CLI (Command Line Interface)

ls - 현재 디렉토리 내용 나열



The screenshot shows a Windows File Explorer window with the address bar displaying the path: > This PC > Documents > 멋쟁이사자처럼 운영진 > 0차 기초강의 > ls sample. The main area displays a list of files with columns for Name, Date modified, Type, and Size. There are five files listed, all of which are Text Documents, each with a size of 0 KB. The file '강연우' is currently selected.

Name	Date modified	Type	Size
강단비	2021-03-06 오후 4:46	Text Document	0 KB
강연우	2021-03-06 오후 4:46	Text Document	0 KB
전, 해주예요	2021-03-06 오후 4:46	Text Document	0 KB
정상윤	2021-03-06 오후 4:46	Text Document	0 KB
정성원	2021-03-06 오후 4:46	Text Document	0 KB

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진 /0차 기초강의 /CLI실습파일 /ls_sample
$ ls
강단비.txt 강연우.txt '전, 해주예요.txt' 정상윤.txt 정성원.txt
```

CLI (Command Line Interface)

ls – 자세하게 알아보기

ls 현재 디렉토리 위치에 내용 나열

ls -a 숨김 파일(.git, .gitignore 등) 포함해서 모두 보여줌

ls -l 좀 더 자세한 정보 (수정 날짜 등) 포함해서 출력

ls -al ls -a, ls -l 둘 다 같이 섞어서 보여줌

CLI (Command Line Interface)

ls - 자세하게 알아보기

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진 /0차 기초강의 /CLI실습파일 /ls_sample
$ ls -al
total 4
drwxr-xr-x 1 User 197609 0 Mar  6 16:57 ./
drwxr-xr-x 1 User 197609 0 Mar  6 16:48 ../
-rw-r--r-- 1 User 197609 0 Mar  6 16:46 강단비.txt
-rw-r--r-- 1 User 197609 0 Mar  6 16:46 강연우.txt
-rw-r--r-- 1 User 197609 0 Mar  6 16:46 '전, 해 주세요.txt'
-rw-r--r-- 1 User 197609 0 Mar  6 16:46 정상운.txt
-rw-r--r-- 1 User 197609 0 Mar  6 16:46 정성원.txt
```

CLI (Command Line Interface)

mkdir - 디렉토리 만들기

mkdir 폴더_이름

폴더_이름 을 가진 폴더(디렉토리)를
생성해요!

```
User@YeongBeom MINGW64 ~/Documents/몇 장 이 사 자 처 럼 운 영 진 /0차 기 초 강 의 /CLI실 습 파
일 /ls_sample
$ mkdir Nice_Folder

User@YeongBeom MINGW64 ~/Documents/몇 장 이 사 자 처 럼 운 영 진 /0차 기 초 강 의 /CLI실 습 파
일 /ls_sample
$ ls
Nice_Folder/  강 단 비 .txt  강 연 우 .txt  '전 , 해 주 예 요 .txt'  정 상 운 .txt  정 성 원 .txt
```

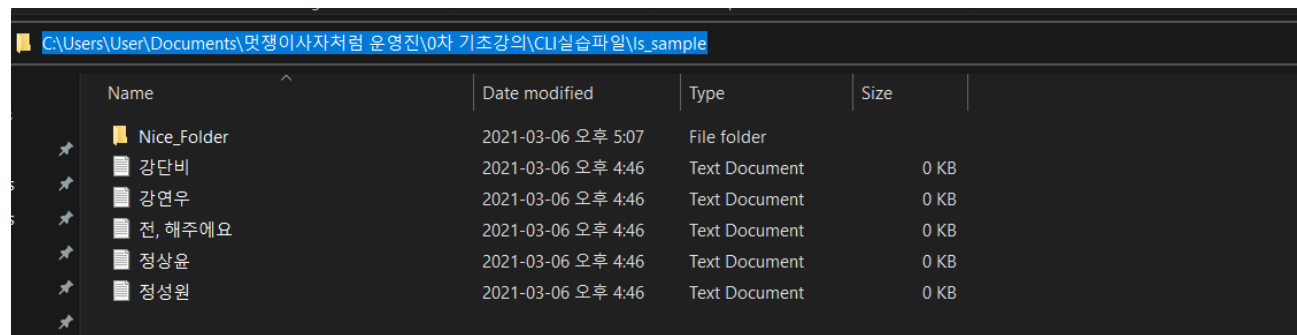
CLI (Command Line Interface)

pwd – 현재 절대 경로 확인

pwd

현재 절대 경로를 확인합니다.

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/CLI실습파일/ls_sample
$ pwd
/c/Users/User/Documents/멋쟁이사자처럼 운영진/0차 기초강의/CLI실습파일/ls_sample
```



Name	Date modified	Type	Size
Nice_Folder	2021-03-06 오후 5:07	File folder	
강단비	2021-03-06 오후 4:46	Text Document	0 KB
강연우	2021-03-06 오후 4:46	Text Document	0 KB
전, 해주예요	2021-03-06 오후 4:46	Text Document	0 KB
정상윤	2021-03-06 오후 4:46	Text Document	0 KB
정성원	2021-03-06 오후 4:46	Text Document	0 KB

CLI (Command Line Interface)

cd – 경로 이동

cd 원하는_경로

(원하는 경로)의 주소값으로 이동해요!

(원하는 경로)는 절대 경로, 상대 경로 둘 다

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/CLI실습파일/ls_sample
$ ls
Nice_Folder/ 강단비.txt 강연우.txt '전, 해주세요.txt' 정상은.txt 정성원.txt
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/CLI실습파일/ls_sample
$ cd Nice_Folder/
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/CLI실습파일/ls_sample/Nice_Folder
$
```


CLI (Command Line Interface)

cd – 경로 이동

cd 원하는_경로

(원하는 경로)의 주소값으로 이동해요!

(원하는 경로)는 절대 경로, 상대 경로 둘 다

????

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/CLI실습파일/ls_sample
$ ls
Nice_Folder/ 강단비.txt 강연우.txt '전, 해주세요.txt' 정상은.txt 정성원.txt
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/CLI실습파일/ls_sample
$ cd Nice_Folder/
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/CLI실습파일/ls_sample/Nice_Folder
$
```

CLI (Command Line Interface)

절대 경로 vs 상대 경로

절대 경로

Root 디렉토리에서 해당 디렉토리까지의 경로 (pwd를 사용했을 때 나오는 경로)

상대 경로

현재 디렉토리에서 해당 디렉토리까지의 경로

" ." : 현재 경로를 의미 (Sample_dir/Sample_son_dir/now_path)

" .. " : 위의 경로를 의미 (Sample_dir/Sample_son_dir)

추가 : <https://blog.naver.com/next/samplenumber23>은 뭘 의미할지 추측해보세요!

CLI (Command Line Interface)

cd – 위의 경로로 이동

cd ..

현재 경로에서 위의 디렉토리로
이동해요!

```
User@YeongBeom MINGW64 ~/Documents/몇 장 이 사 자 저 렴 은 영 진 /0차 기 초 강 의 /CLI실 습 과 일 /ls_sample/Nice_Folder
$ cd ..
User@YeongBeom MINGW64 ~/Documents/몇 장 이 사 자 저 렴 은 영 진 /0차 기 초 강 의 /CLI실 습 과 일 /ls_sample
$ |
```

CLI (Command Line Interface)

touch - 파일 생성

touch (생성하려는 이름)

현재 디렉터리에서 (생성하려는 이름)을
가진 파일을 생성해요!

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0자 기초강의/CLI실습파일/ls_sample
$ ls
강단비.txt  강연우.txt  '전, 해주세요.txt'  정상운.txt  정성원.txt

User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0자 기초강의/CLI실습파일/ls_sample
$ touch README.md

User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0자 기초강의/CLI실습파일/ls_sample
$ ls
README.md  강단비.txt  강연우.txt  '전, 해주세요.txt'  정상운.txt  정성원.txt
```

CLI (Command Line Interface)

rm - 삭제

rm (삭제_파일_이름)

현재 디렉터리에서 (생성하려는 이름)을 가진 파일을 삭제해요!

rm -rf (삭제_폴더_이름)

현재 디렉터리에서 (삭제_폴더_이름)을 가진 폴더를 삭제해요!

```
User@YeongBeom MINGW64 ~/Documents/몇몇이사자처럼운영진/0차기초강의/CLI실습파일/ls_sample
$ ls
Nice_Folder/  README.md  강단비.txt  강연우.txt  '전, 해주세요.txt'  정상운.txt  정성원.txt

User@YeongBeom MINGW64 ~/Documents/몇몇이사자처럼운영진/0차기초강의/CLI실습파일/ls_sample
$ rm README.md

User@YeongBeom MINGW64 ~/Documents/몇몇이사자처럼운영진/0차기초강의/CLI실습파일/ls_sample
$ ls
Nice_Folder/  강단비.txt  강연우.txt  '전, 해주세요.txt'  정상운.txt  정성원.txt

User@YeongBeom MINGW64 ~/Documents/몇몇이사자처럼운영진/0차기초강의/CLI실습파일/ls_sample
$ rm -rf Nice_Folder/

User@YeongBeom MINGW64 ~/Documents/몇몇이사자처럼운영진/0차기초강의/CLI실습파일/ls_sample
$ ls
강단비.txt  강연우.txt  '전, 해주세요.txt'  정상운.txt  정성원.txt

User@YeongBeom MINGW64 ~/Documents/몇몇이사자처럼운영진/0차기초강의/CLI실습파일/ls_sample
$
```

Extra Curriculum CLI - Vim









가장 널리 쓰이는 텍스트 에디터 중 하나

- 단축키가 편리한 게 많음
- 터미널에서 바로 접속하여 사용가능
- 사양 낮은 컴퓨터 및 리눅스에서도 기본 설치됨
- 배우기가 힘들.

공부하다 궁금한 거 있으면 허영범 / 최주원에게









Git?

과제 할 때, 생겨나는 버전들

 멋쟁이사자처럼 기획안 허영범	2021-03-06 오후 8:23
 멋쟁이사자처럼 기획안 허영범2	2021-03-06 오후 8:24
 멋쟁이사자처럼 기획안 허영범수정완료	2021-03-06 오후 8:24
 멋쟁이사자처럼 기획안 허영범 최종본	2021-03-06 오후 8:24
 멋쟁이사자처럼 기획안 허영범 수정	2021-03-06 오후 8:24
 멋쟁이사자처럼 기획안 허영범 최종본 진짜진짜마지막	2021-03-06 오후 8:24

Git?

협업까지 해버리면 미쳐버리죠.

 멋사 교안 강연우	2021-03-06 오후 8:23	서식 있는 텍스트	1 KB
 멋사 교안 류지현	2021-03-06 오후 8:24	서식 있는 텍스트	39 KB
 멋사 교안 종합 정성원빼고	2021-03-06 오후 8:24	서식 있는 텍스트	39 KB
 멋사 교안 이소영	2021-03-06 오후 8:24	서식 있는 텍스트	39 KB
 멋사 교안 종합 허영범 보충	2021-03-06 오후 8:24	서식 있는 텍스트	39 KB
 멋사 교안 종합 허영범 사람인가	2021-03-06 오후 8:24	서식 있는 텍스트	39 KB
 멋사 교안 허영범	2021-03-06 오후 8:24	서식 있는 텍스트	39 KB
 멋사 교안 이한주	2021-03-06 오후 8:24	서식 있는 텍스트	39 KB

| Git?



분산형 버전 관리 시스템

- 버전 관리
- 협업 지원

| Git?



분산형 버전 관리 시스템

- ✓ - 버전 관리
- 협업 지원 (나중 5월 세션 때 진행)

| Git 초기 세팅

`git config --global user.name "이름/닉네임"`

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

Git Repository 초기화

아무것도 안했을 때 초기 상태

Local

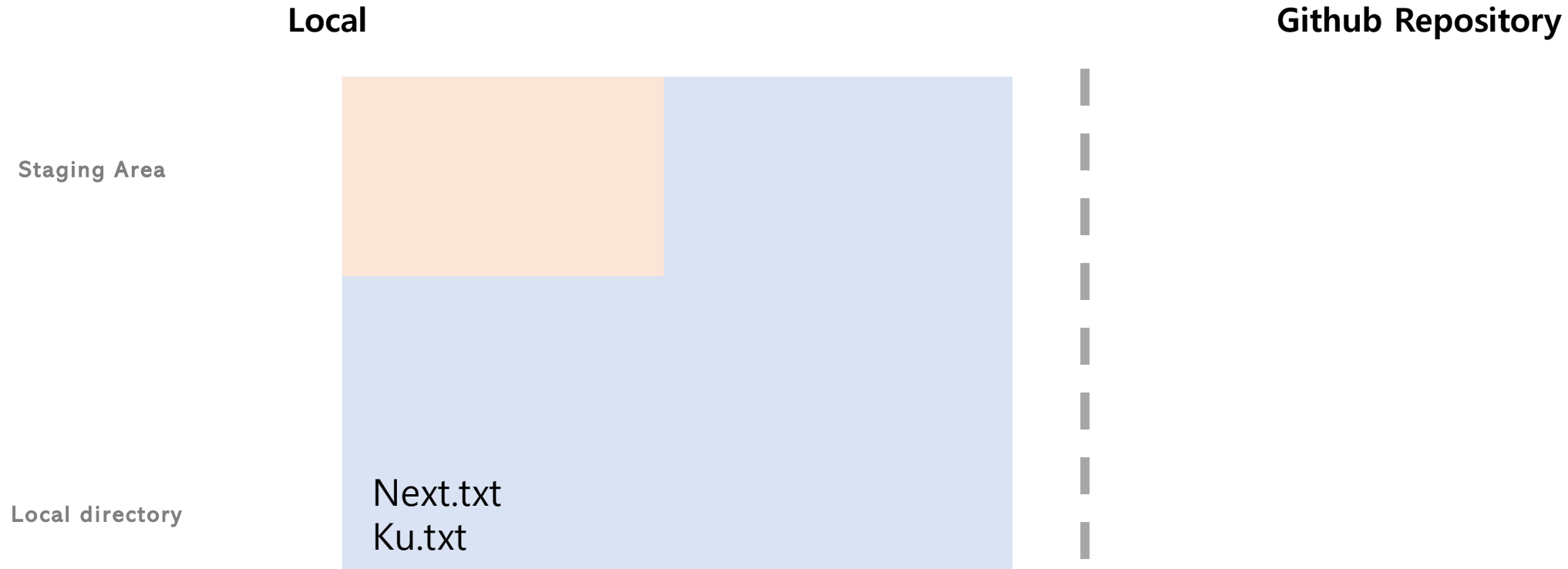
Github Repository

Local directory

Next.txt
Ku.txt

Git Repository 초기화

Git 저장소 활성화



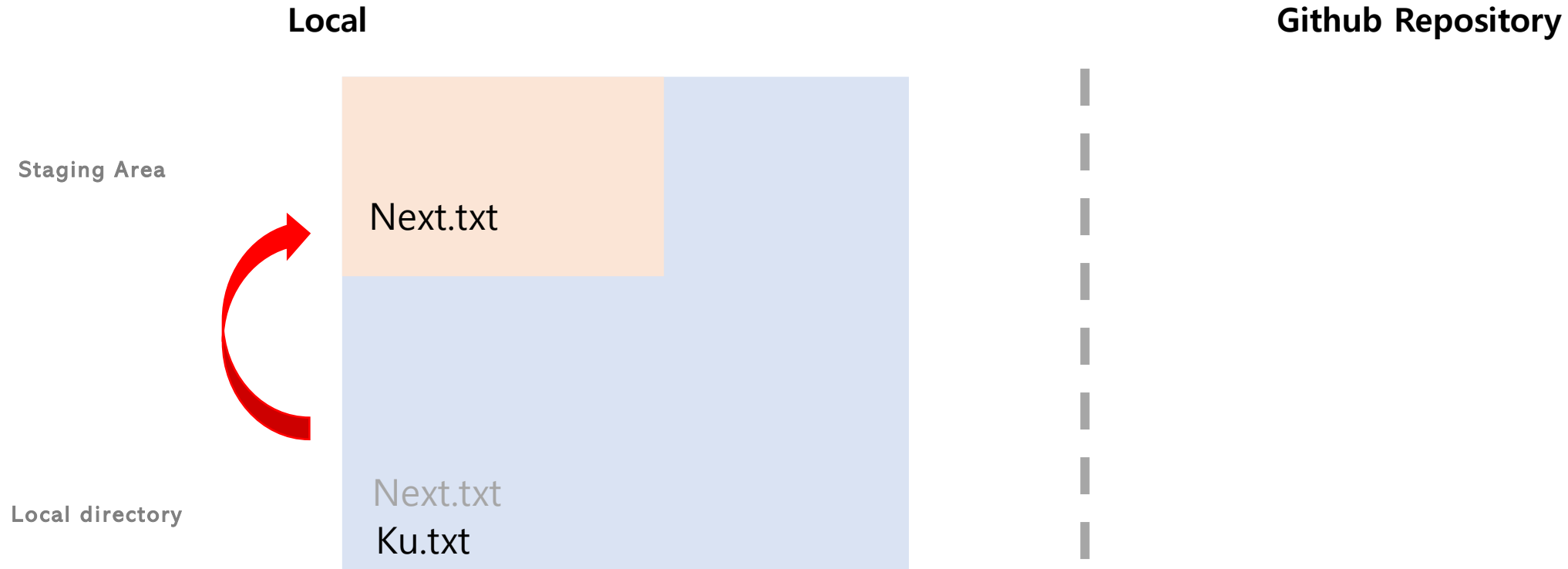
git init

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/git_session
$ git init
Initialized empty Git repository in C:/Users/User/Documents/멋쟁이사자처럼 운영진/0차 기초강의/git_session/.git/

User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/git_session (master)
$ |
```

Git 스테이지에 파일 추가

해당 명령어를 통해서 바뀐 상황들을 git에 스냅샷으로 남겨요



git add (파일이름)

git add Next.txt

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/git_session (master)
$ git add Next.txt
```

Git 추적 상황 확인

해당 명령어를 통해서 현재 git이 어떤 상황인지 파악할 수 있습니다.

Local

Github Repository

Staging Area

Next.txt

Local directory

Ku.txt

git status

Next.txt 추적 중,
Ku.txt 추적 아님

```
User@YeongBeom MINGW64 ~/Documents/몇 정 이 사 자 처 럼 온 영 진 /0차 기초 강의 /git_session (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Next.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Ku.txt
```

Git 커밋

해당 명령어를 통해서 지금까지 상황을 저장할 수가 있어요.



git commit -m '(커밋메세지)'

git commit -m "첫커밋"

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼_운영진/0차_기초캠프/git_session (master)
$ git commit -m "첫 커밋"
[master (root-commit) c7df116] 첫 커밋
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Next.txt
```


Git 기록 확인

이때까지의 커밋 내역을 확인해보아요.



git log

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/git_session (master)
$ git log
commit c7df11614b38965a2ea157a379e5f5cdead71651 (HEAD -> master)
Author: hybeom0720 <hybeom@gmail.com>
Date: Sat Mar 6 21:06:57 2021 +0900

첫 커밋
```

Git 파일 전부 추적

아까 배웠던 상대 경로를 활용하여 지금 경로의 것들을 다 추가해보아요.



`git add .`
`git commit -m "두번째 커밋"`

현재 경로를 의미

```
User@YeongBeom MINGW64 ~/Documents/몇정이사자처럼 운영진/0차 기초강의/git_session (master)
$ git add .

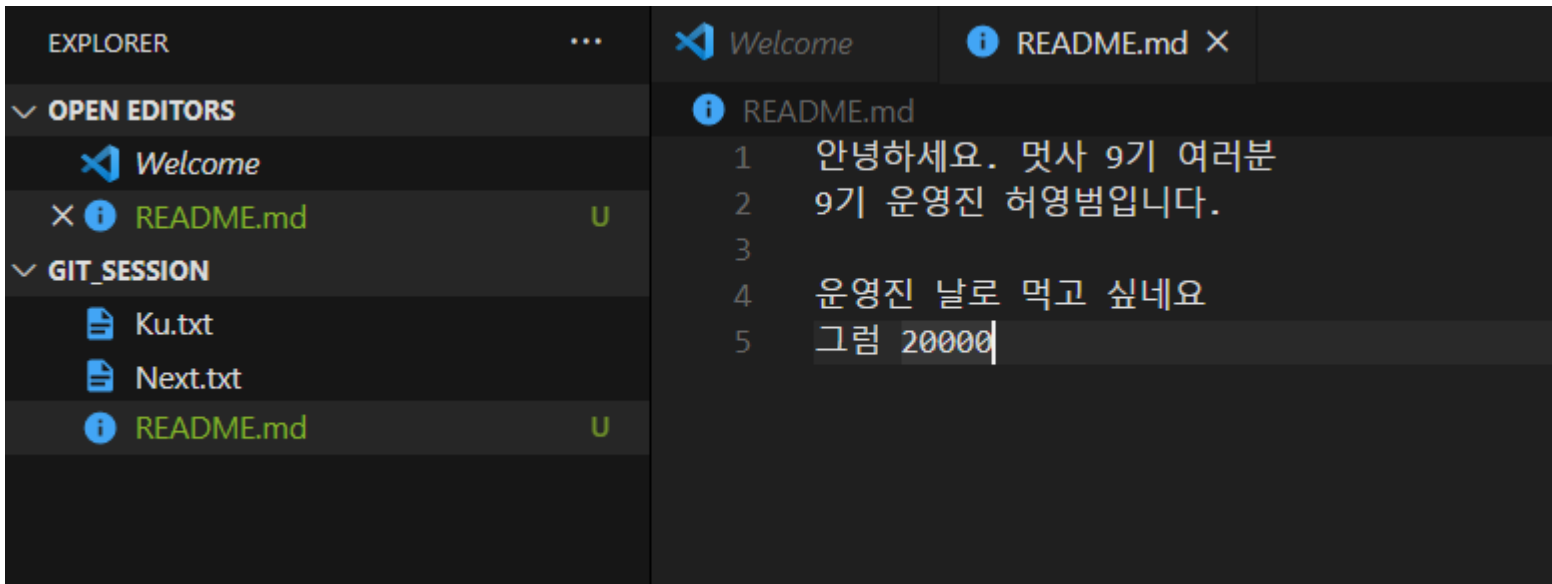
User@YeongBeom MINGW64 ~/Documents/몇정이사자처럼 운영진/0차 기초강의/git_session (master)
$ git commit -m "두번째 커밋"
[master c0cde79] 두번째 커밋
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Ku.txt
```

Github에 푸시하기

README.md를 작성해봅시다.

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/git_session (master)
$ code .
```

code . 해당 경로에서, (vs)code를 열어라!

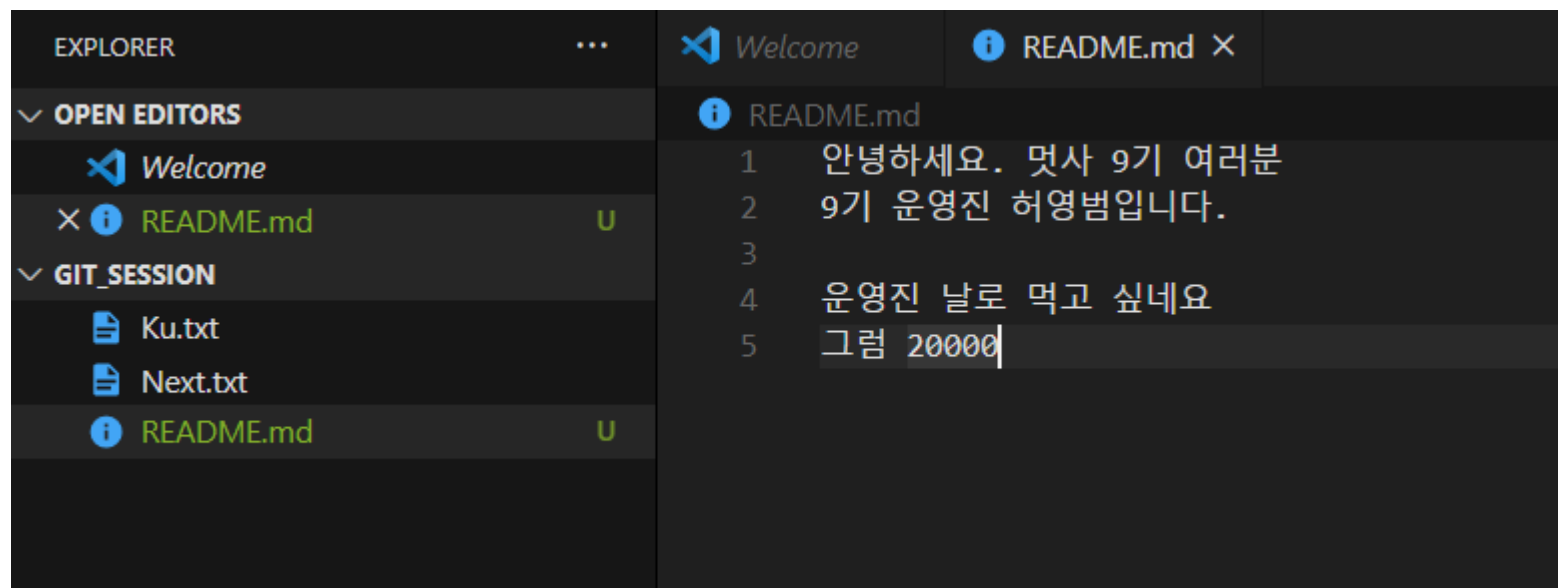


Github란?

README.md를 작성해봅시다.

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진 /0차 기초강의 /git_session (master)
$ code .
```

code . 해당 경로에서, (vs)code를 열어라!



README.md

Repository에서 메인으로 보여질 문서입니다.
사용 설명서나 개발자들이 기록하고 싶은 걸 기록해요.

| Github란?



Git으로 버전 관리되는 코드들의 저장소

- 협업 지원
- 오픈소스의 성지
- 코드 관리 페이지

Github Repository 현 상황

만들어져 있는데... 연결이 안됐네?

Local

Github Repository

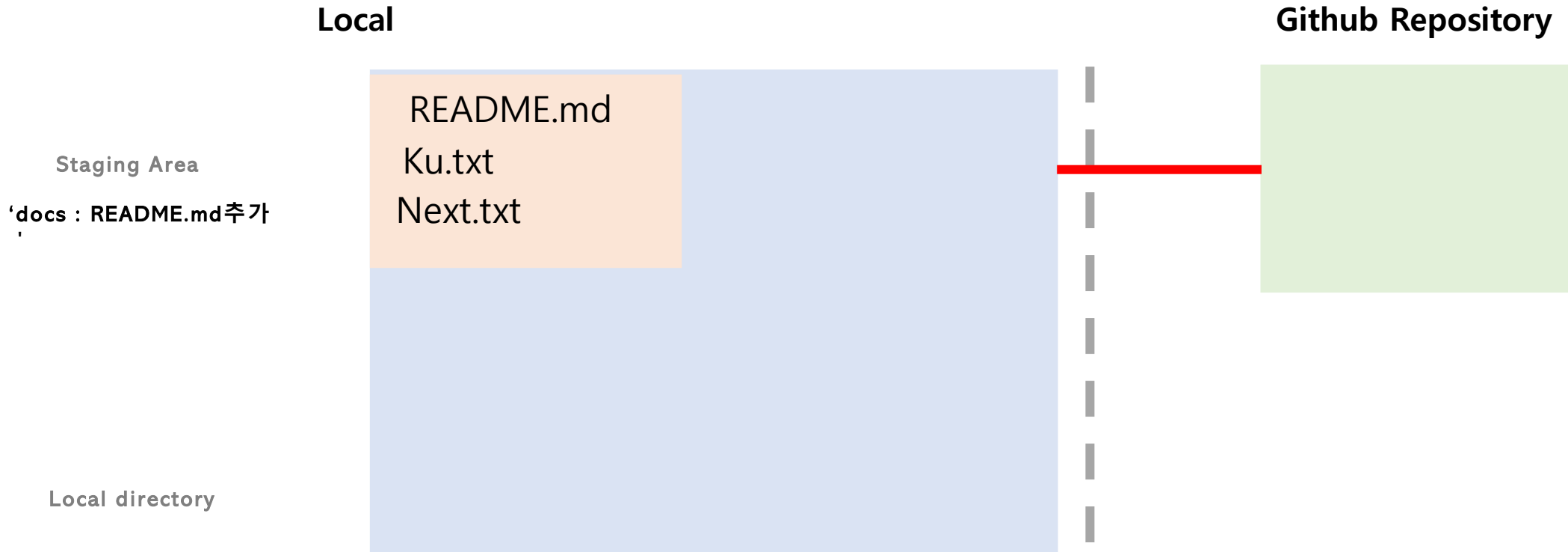
Staging Area
'docs : README.md 추가
,

README.md
Ku.txt
Next.txt

Local directory

Local Repository와 Github Repository 연결

Local Repository와 Github Repository를 연결해봅시다.



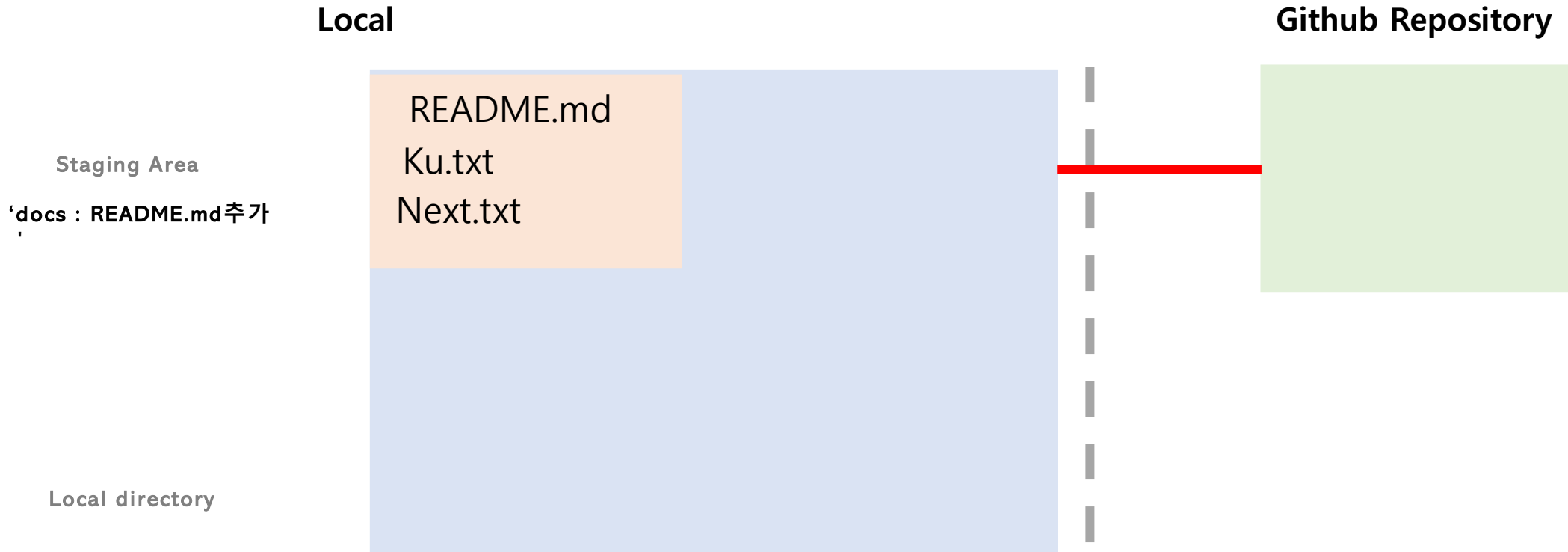
`git remote add "우리가 부를 이름" "Repository 주소"`

ex) `git remote add origin https://github.com/OrcaLion/NEXT_FirstRepository.git`

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼_운영진/0차_기초강의/git_session (master)
$ git remote add origin https://github.com/OrcaLion/NEXT_FirstRepository.git
```

Local Repository와 Github Repository 연결 확인

잘 되어있는거 맞나요?

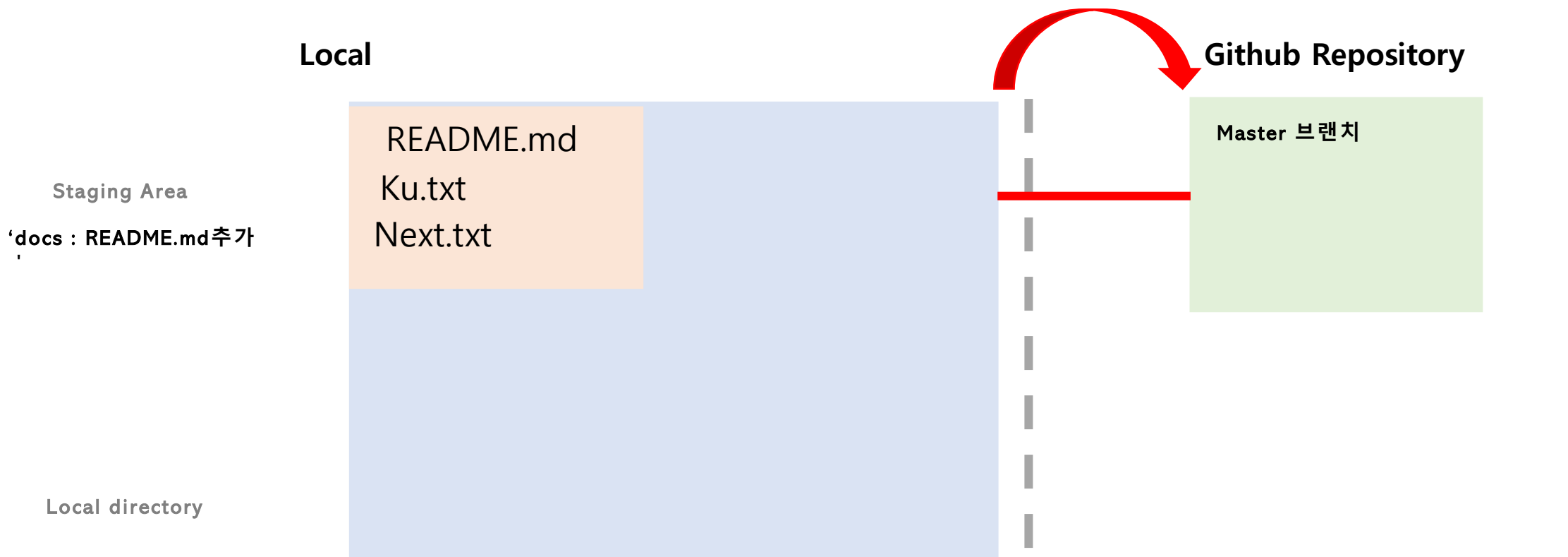


git remote -v

```
User@YeongBeom MINGW64 ~/Documents/멋쟁이사자처럼 운영진/0차 기초강의/git_session (master)
$ git remote -v
origin  https://github.com/OrcaLion/NEXT_FirstRepository.git (fetch)
origin  https://github.com/OrcaLion/NEXT_FirstRepository.git (push)
```


Github Repository에 푸시

인터넷에 내가 짠 코드 한번 올려봅시다






다음 강의 떡밥

`git push "밀 주소 명칭" "밀 브랜치"`
Ex) `git push origin master`


Github 계정과 비밀번호를 입력해주세요


Github Repository에 푸시




인터넷에 내가 짠 코드 한번 올려봅시다


 master ▾  1 branch  0 tags

[Go to file](#) [Add file ▾](#) [↓ Code ▾](#)

 **hybeom0720** docs : README.md 추가

6b7312d 16 minutes ago  3 commits

 Ku.txt	두번째 커밋	18 minutes ago
 Next.txt	첫커밋	25 minutes ago
 README.md	docs : README.md 추가	16 minutes ago

README.md 

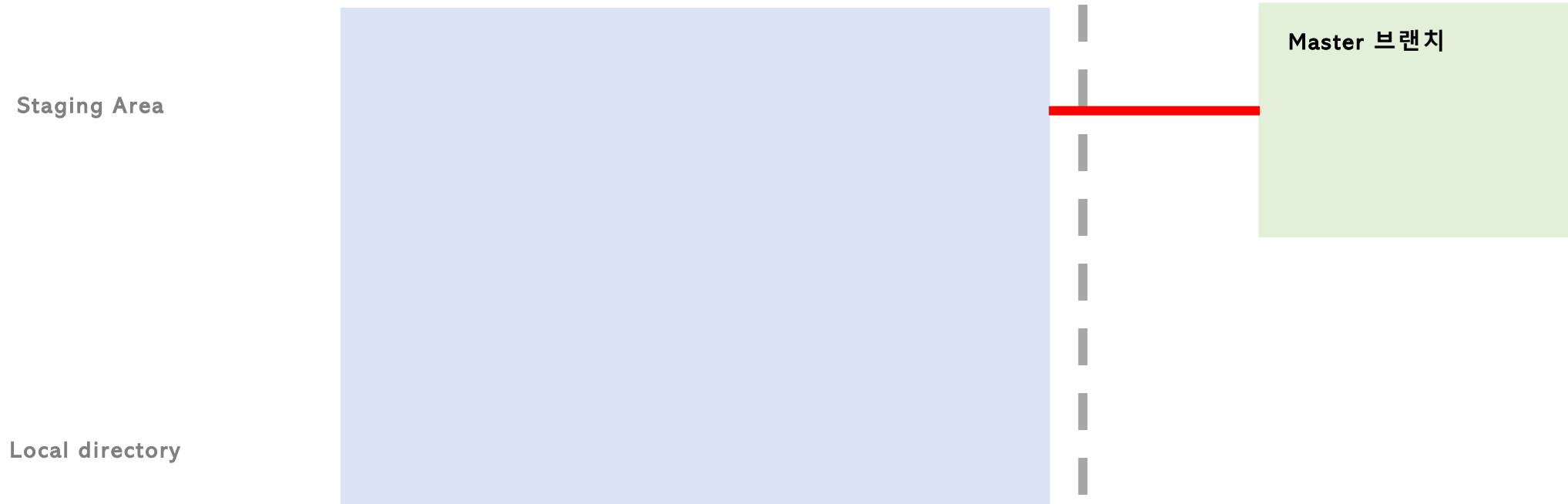
안녕하세요. 멋사 9기 여러분 9기 운영진 허영범입니다.

운영진 날로 먹고 싶네요 그럼 20000

Github Repository에서 코드 당겨오기

밀 수 있으면 받을 수도 있어야겠죠?

기존 폴더와 다른 폴더에서 추가해주세요

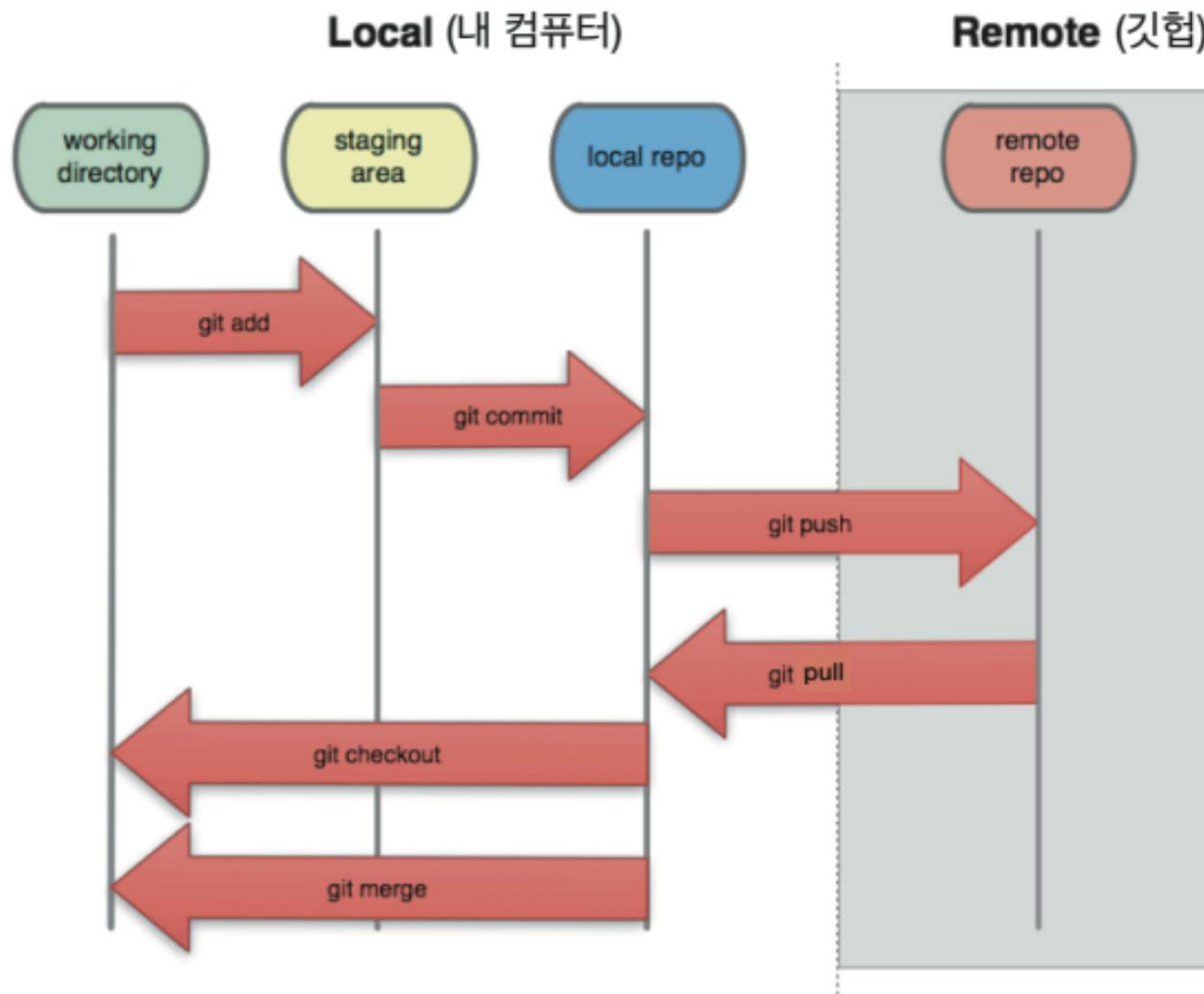


`git clone` "복사해올 깃 허브 주소 "

Ex) `git clone http://github.com/OrcaLion/NEXT_FirstRepository.git`

Git 기능에 대한 그림

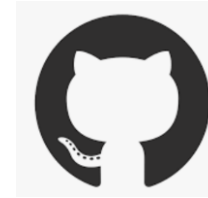
개념 한방 정리.



Detail Curriculum

Git and Github

1. Git 로컬에서 기초 버전관리
2. Github Repository에 푸시하기
3. Git으로 버전 세부적으로 관리 해보기
4. Github 및 Git을 통한 협업



5월 협업 세션에서 진행할 예정입니다!