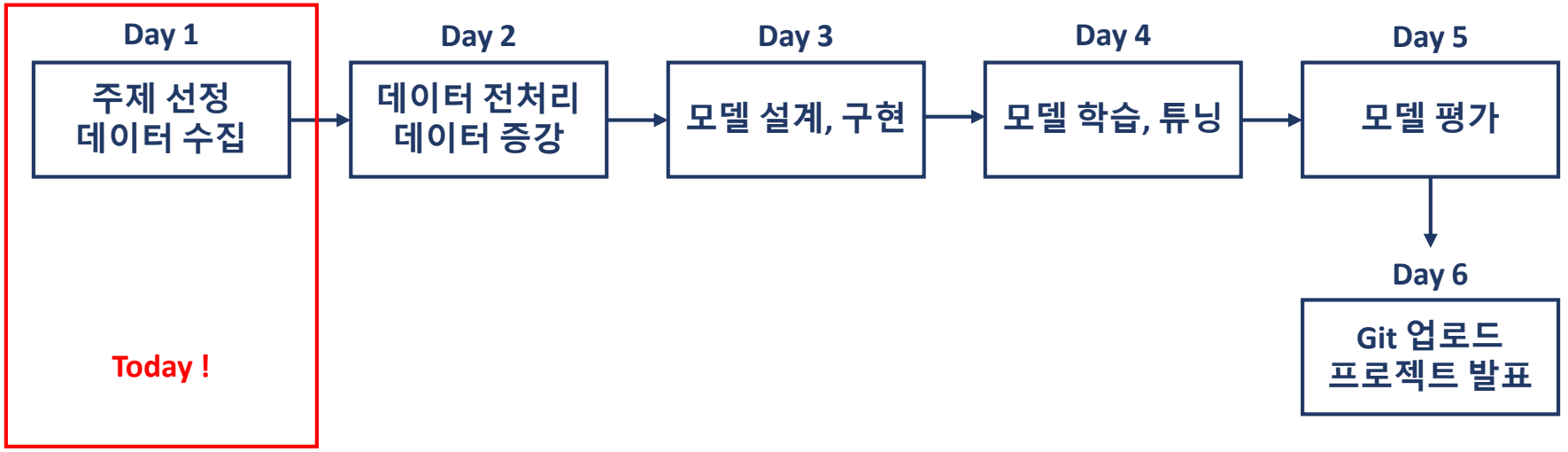
AI 기반 영상 데이터 분석 실습
- Day 1 -

index


Day 1
주제 선정
데이터 수집
Today !
Day 2
데이터 전처리
데이터 증강
Day 3
모델 설계, 구현
Day 4
모델 학습, 튜닝
Day 5
모델 평가
Day 6
Git 업로드
프로젝트 발표

index

1. Git & Github

2. classification 주제 선정

- 어떤 대상을 분류해보고 싶은지?

3. 데이터 수집

Git & Github

index

Git & Github : 주요 기능

Git : 기본 개념

Git 사용법 :

- 설치
- 기초 명령어
- 프로젝트 관리 명령어
- 명령어

Github 사용법 :

- Local to repositories
- Repositories to local

Git & Github : 주요 기능

주요 기능 : git

- 프로젝트 내 문서 수정 이력 관리 가능(특정 시점으로 되돌리기 등)
- 하나의 프로젝트를 여러 사람이 함께 수행할 때 각자 개발 버전 관리 가능

주요 기능 : github

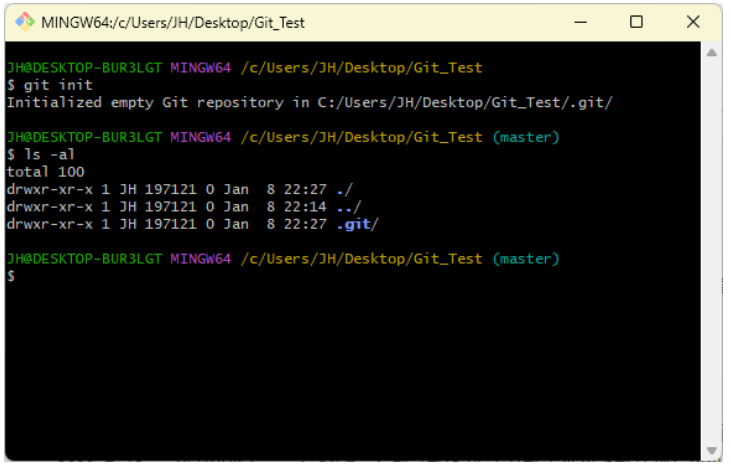
- 깃 저장소 호스팅 지원
- 온라인 상에 저장소 만들기 가능

Git : 기본 개념

주로 사용하는 인터페이스 (실습 시 사용)

: CLI (Command – Line Iterface)

- 깃만 설치하면 바로 사용할 수 있는 기본적인 방법
- 깃의 모든 기능을 지원하는 인터페이스

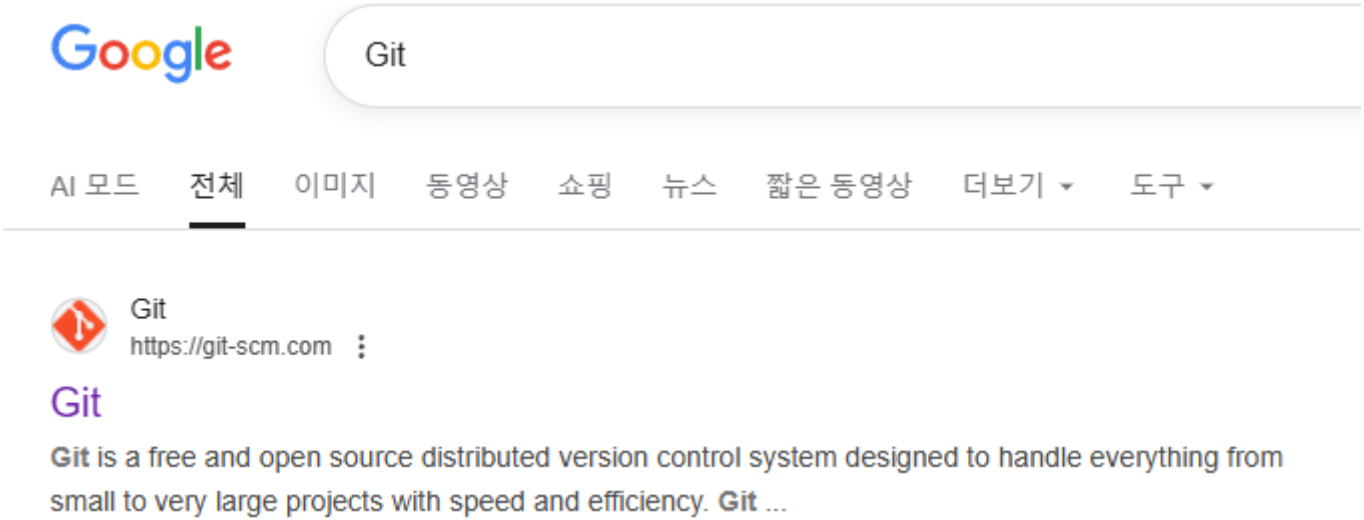


Git의 세가지 관리영역

1. Working Directory (이하 WD) :
   프로젝트 폴더 (작업 하는 폴더)
2. Staging Area (이하 SA) :
   수정 이력 기록할 파일 대기 장소
3. Repository (이하 R):
   staging area에서 대기중인 파일이 최종적으로 기록(커밋)되는 영역

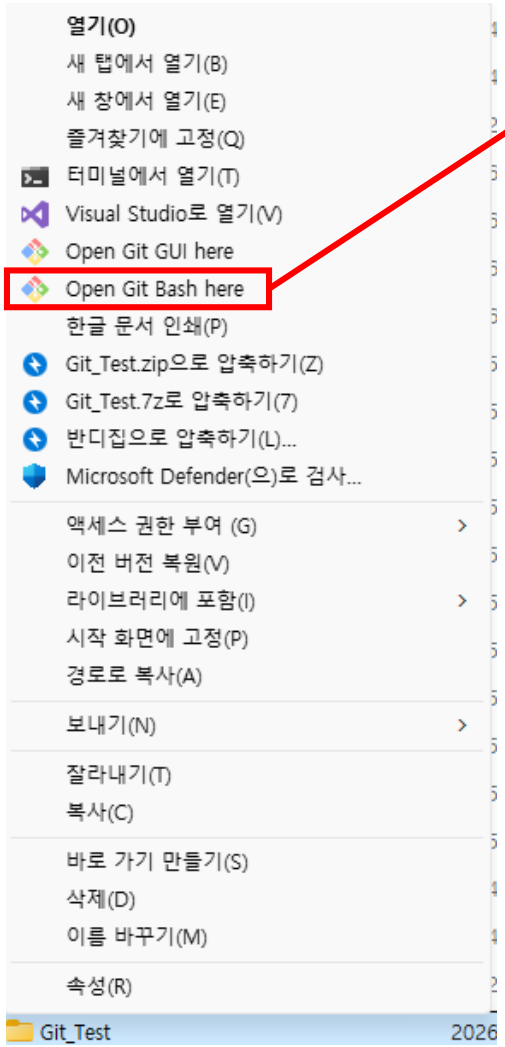
Git 사용법 : 설치

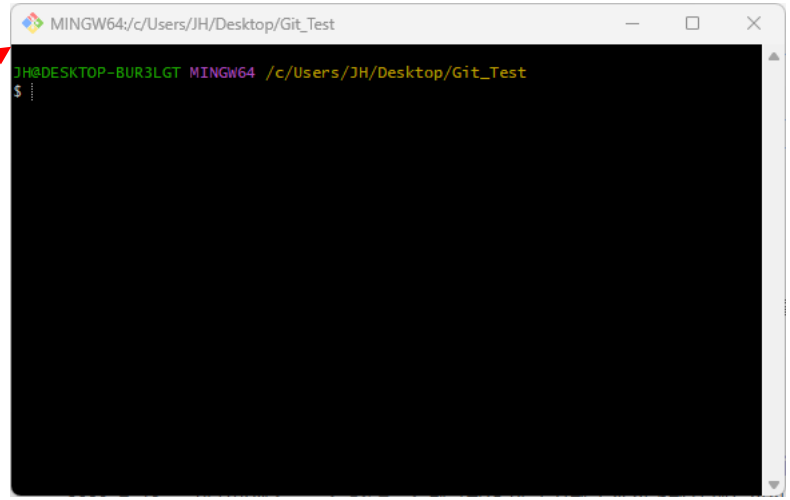
1. Git 설치

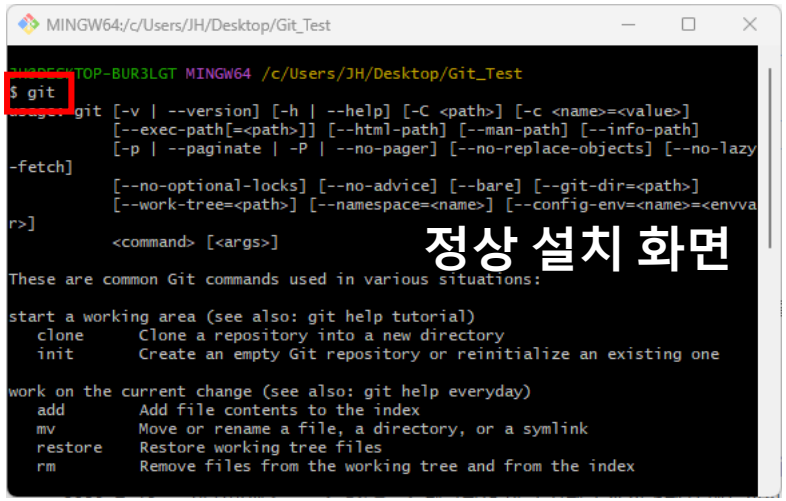


Git 사용법 : 설치

2. 터미널 열기 : 깃 또는 시스템 명령어 사용 가능 & 설치 확인





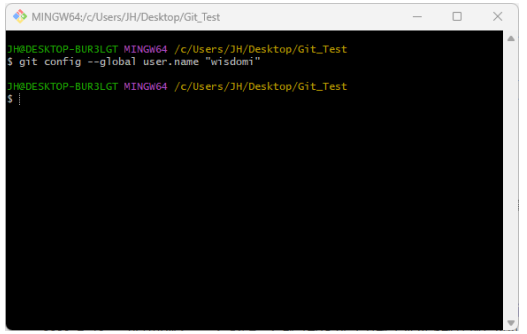


Git 사용법 : 기초 명령어

3. 기초 명령어

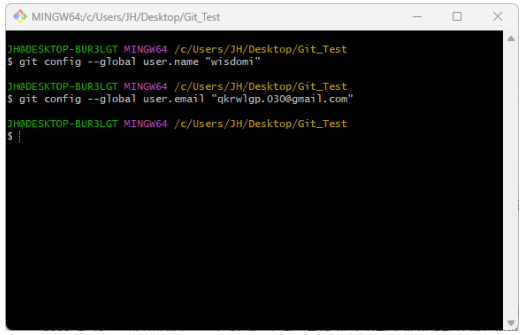
- $git config –global user.name “사용자명”

  -> 사용자명 설정



- $git config –global user.email

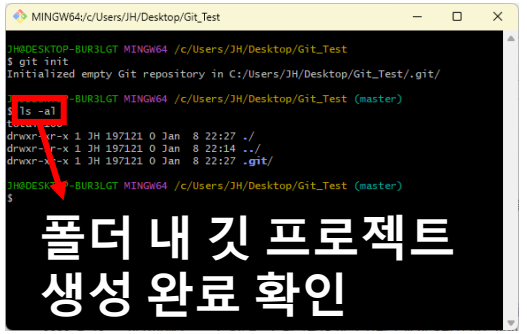
  -> 사용자 메일 주소 설정



- $git init

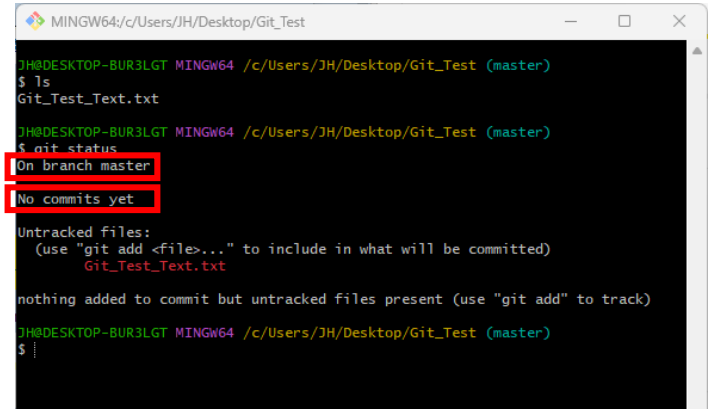
  -> 터미널이 열려있는 폴더를 깃 저장소로 초기화하는 깃 명령어

  (이 폴더를 깃에서 관리하겠다)

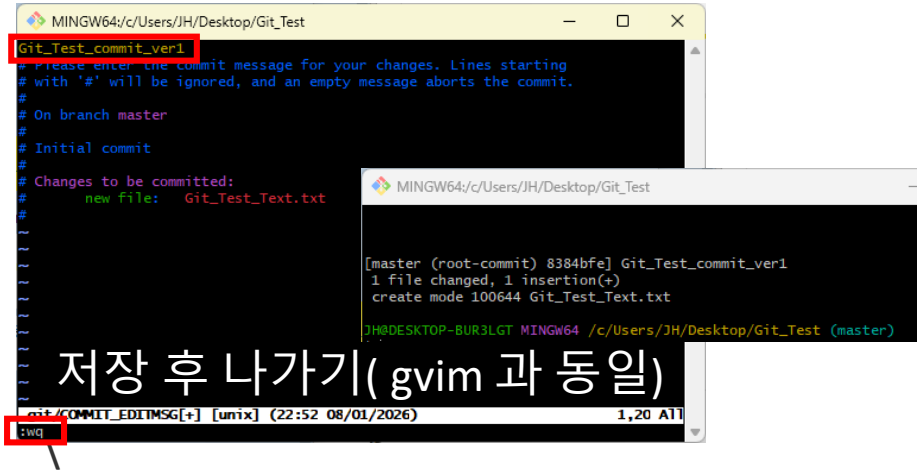


Git 사용법 : 프로젝트 관리 명령어

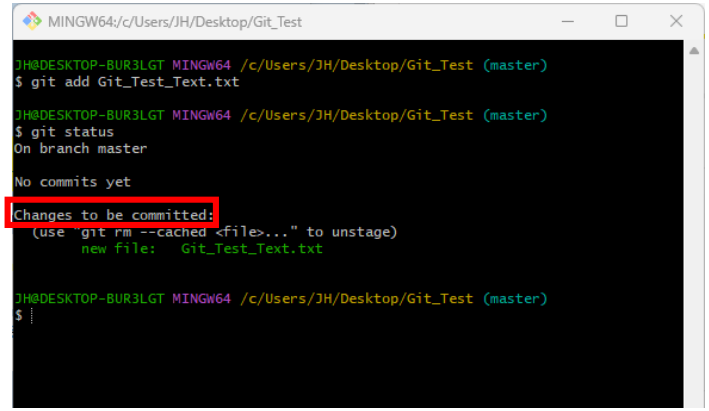
- $git status
  -> 깃 프로젝트 상태 확인



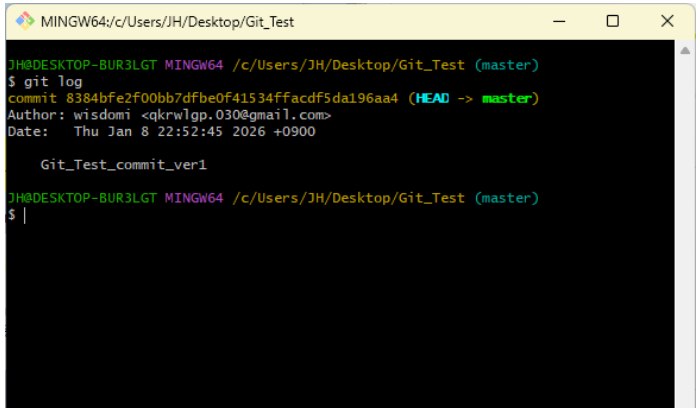
- $git commit
  -> SA영역 문서를 R에 추가 (커밋)



- $git add
  -> WD영역 문서를 SA에 추가



- $git log
  -> 커밋한 수정 이력 확인



Git 사용법 : 명령어

6. 명령어

- $git log -p

  ->커밋 변경 사항까지 함께 출력

- $git log –oneline

  ->각 커밋 요약해 한줄씩 출력

- $git checkout 커밋해시

  ->문서 내용을 특정 커밋으로 되돌림

- $git reflog

  ->HEAD 포인터 참조 이력 출력

Git 사용법 : 명령어

6. 명령어

- $git log -p

  ->커밋 변경 사항까지 함께 출력

- $git log –oneline

  ->각 커밋 요약해 한줄씩 출력

- $git checkout 커밋해시12jknek21jh

  ->문서 내용을 특정 커밋으로 되돌림

- $git reflog

  ->HEAD 포인터 참조 이력 출력

- $git reset

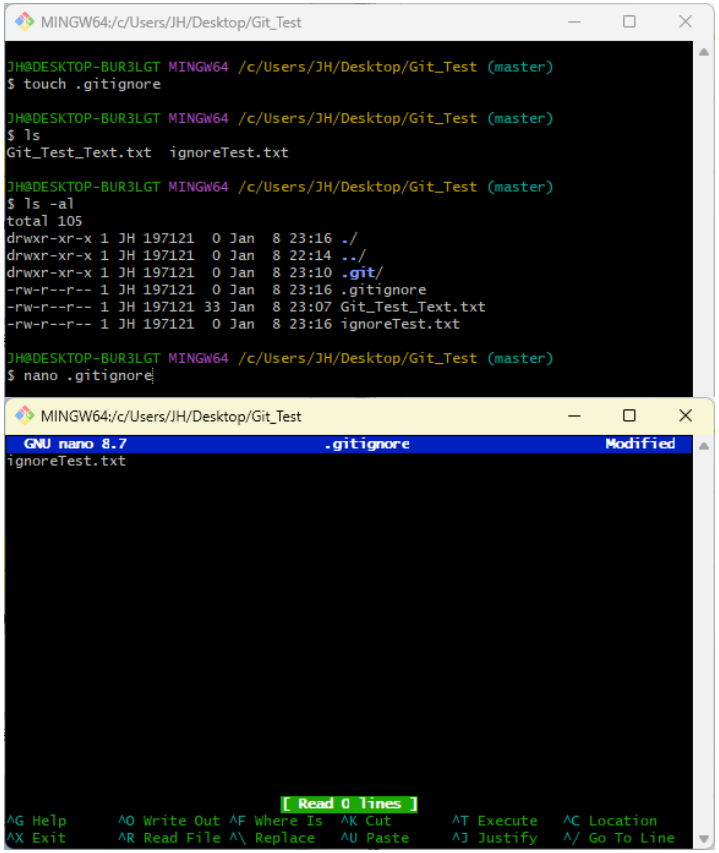
  -> commit 이력 되돌리기

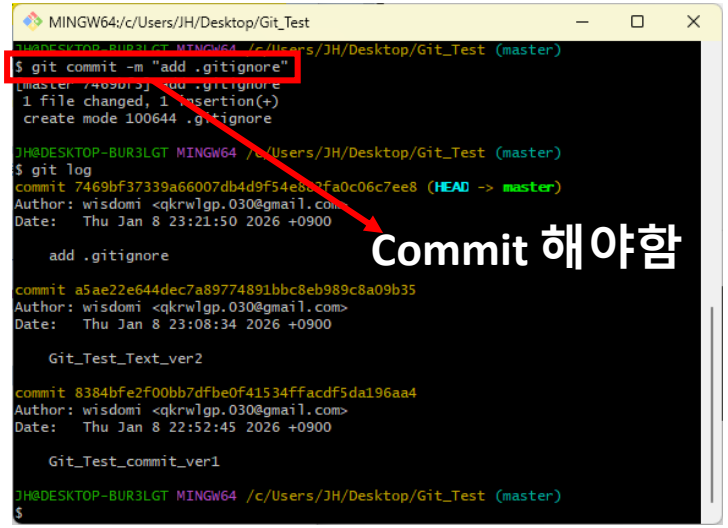
Git 사용법 : 명령어

6. 명령어

- $git ignore

  ->수정 이력에서 제외, 수정 사항이 있어도 감지 안함





Github 사용법 : Local to repositories

1. **Local - repositories**

- $git remote -v

  -> 현재 깃 프로젝트에 등록된 원격 저장소 확인

- $git remote add 원격저장소이름 원격저장소주소

  ->현재 깃 프로젝트에 원격 저장소 등록 & 이름 붙이기

- $git push

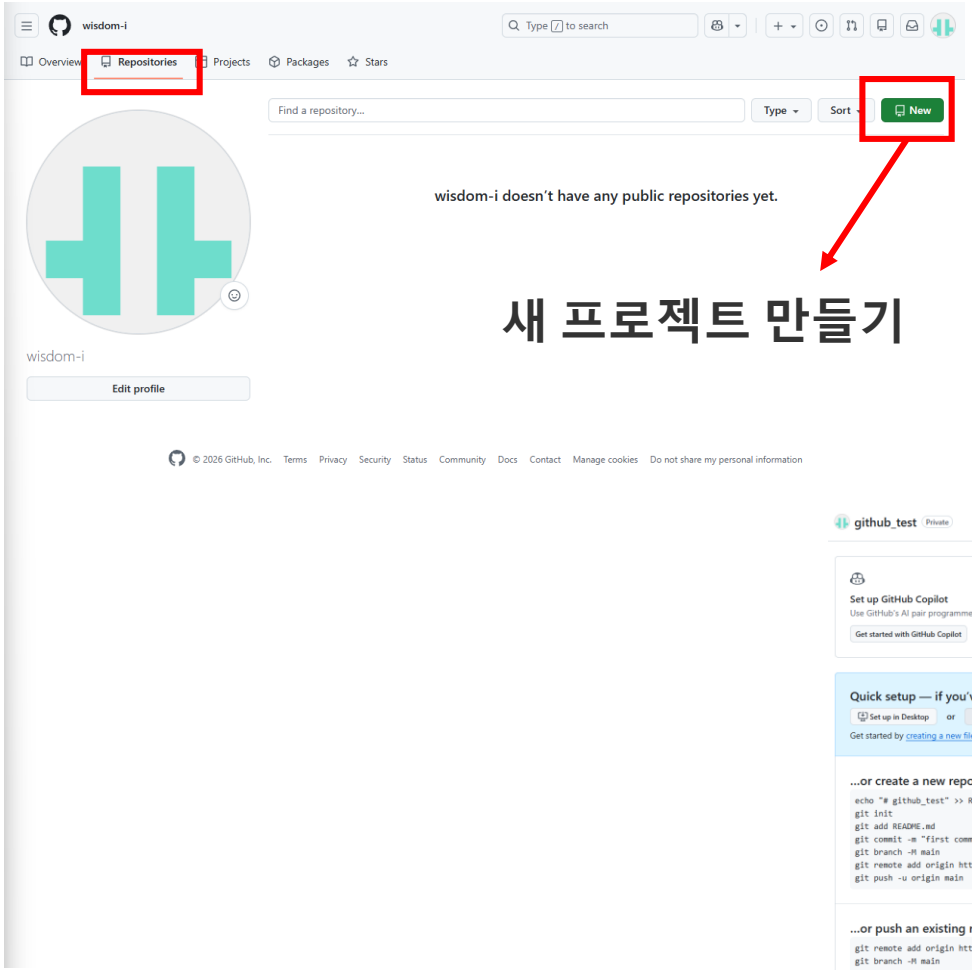
  ->로컬 저장소 내용을 원격 저장소에 공유

- $git pull

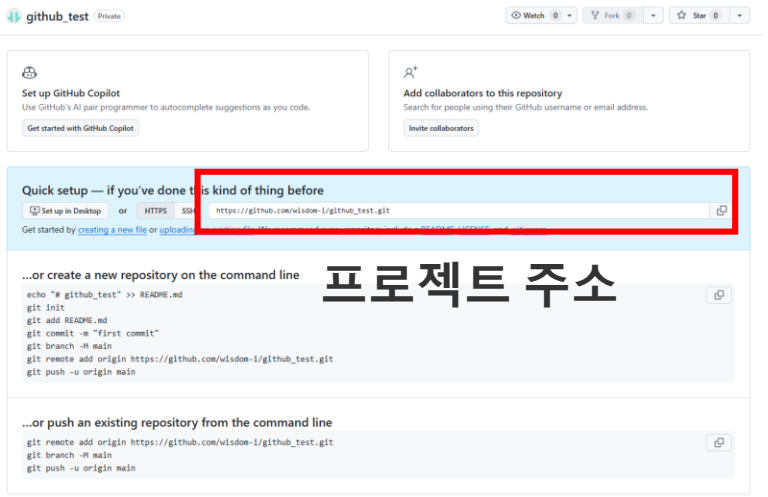
  -> 원격 저장소 내용을 로컬 저장소에 공유

Github 사용법 : Local to repositories

1. Local - repositories



만든 후 안내 페이지

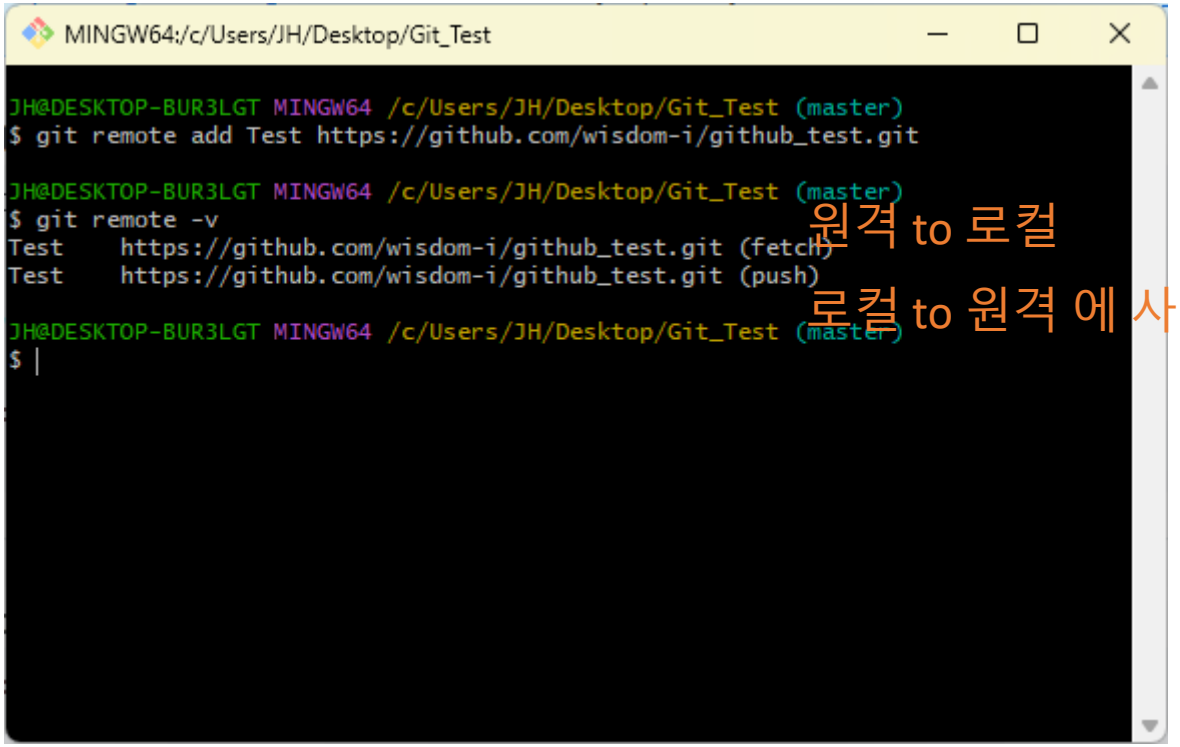


Github 사용법 : Local to repositories

2. Local – repositories : 원격 저장소와 연결

$git remote add 이름지정 주소

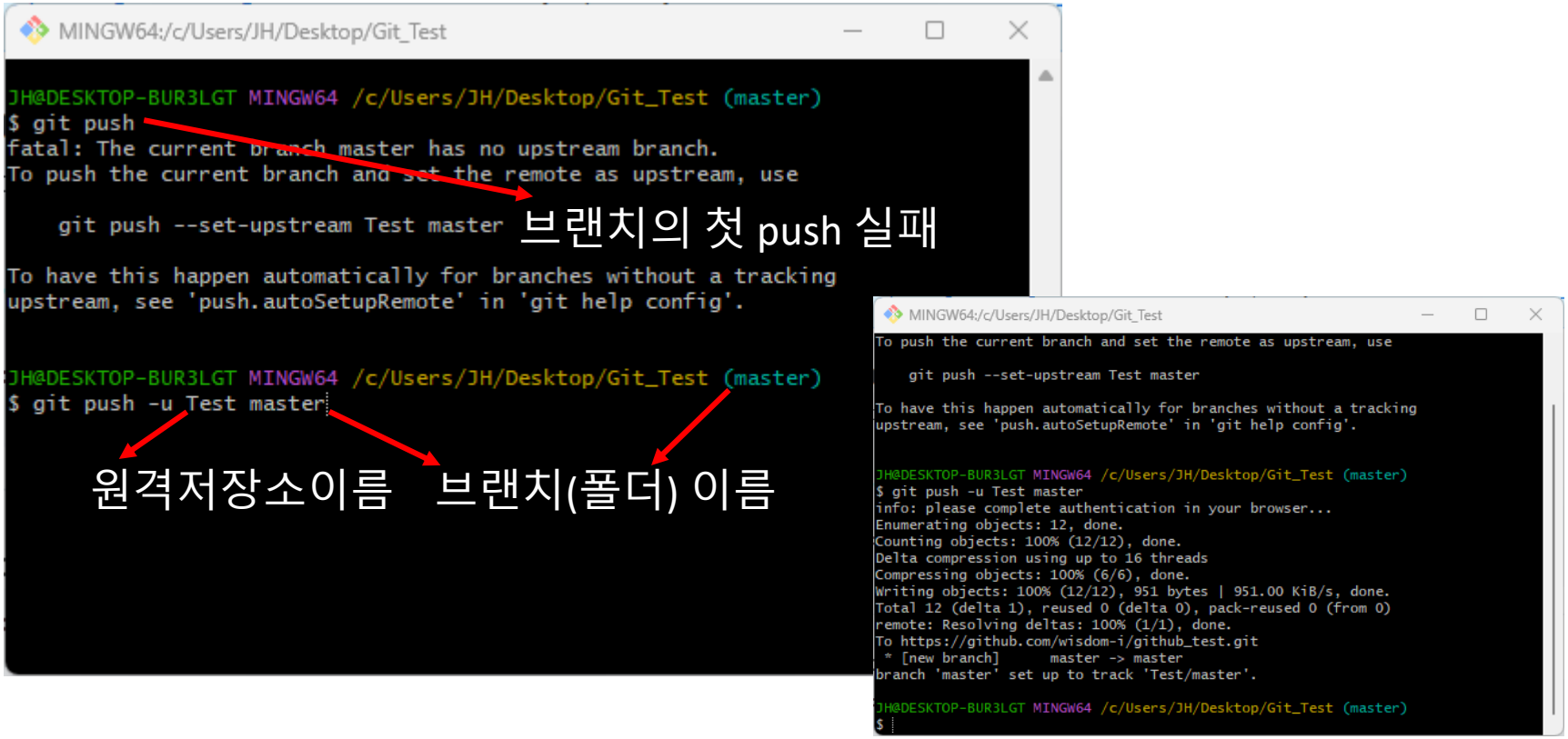
-> 원격 저장소와 연결



Github 사용법 : Local to repositories

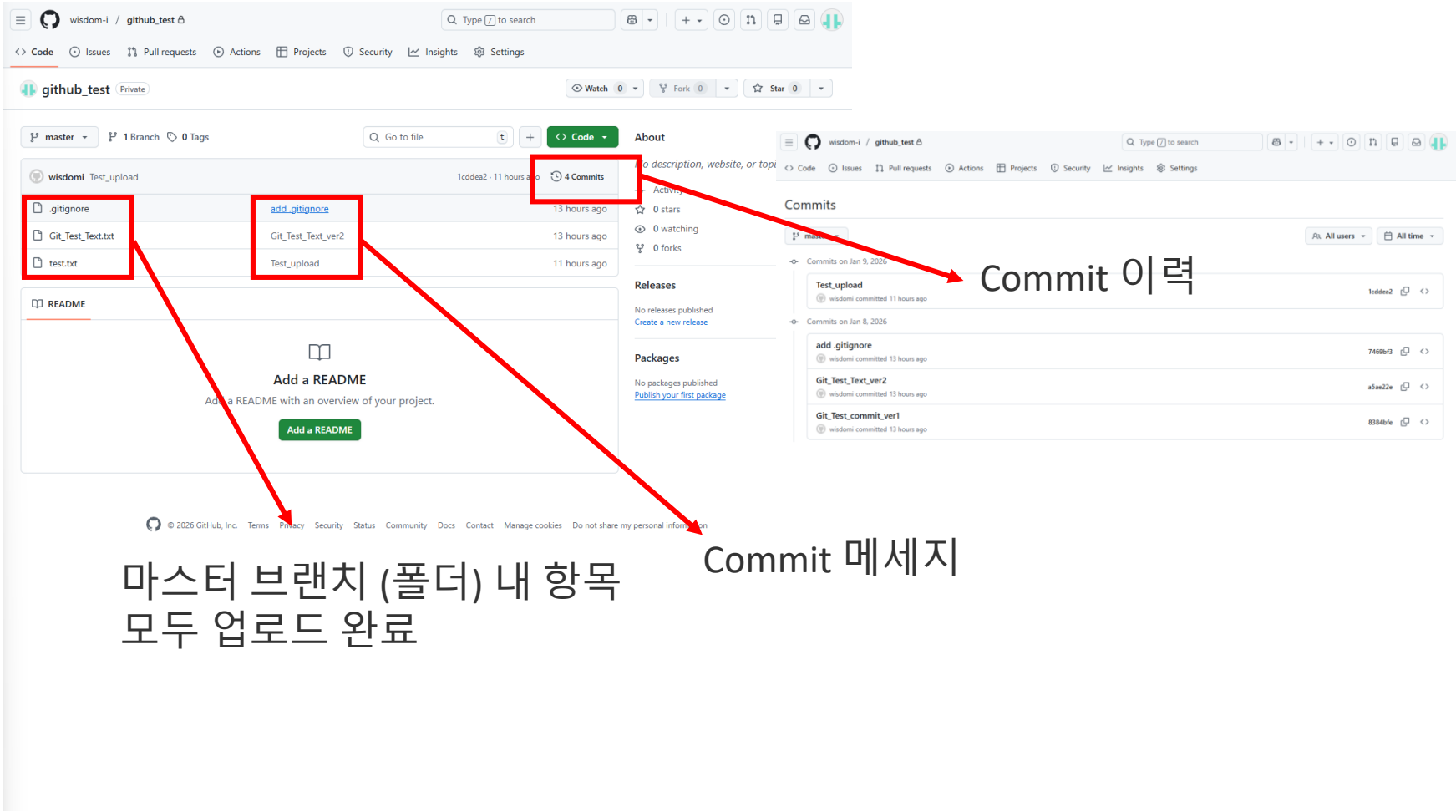
2. Local – repositories : 원격 저장소와 연결

- **$git push –u** 원격저장소이름 브랜치이름
- 이후 같은 브랜치에서 **$git push**로만 사용 가능



Github 사용법 : Local to repositories

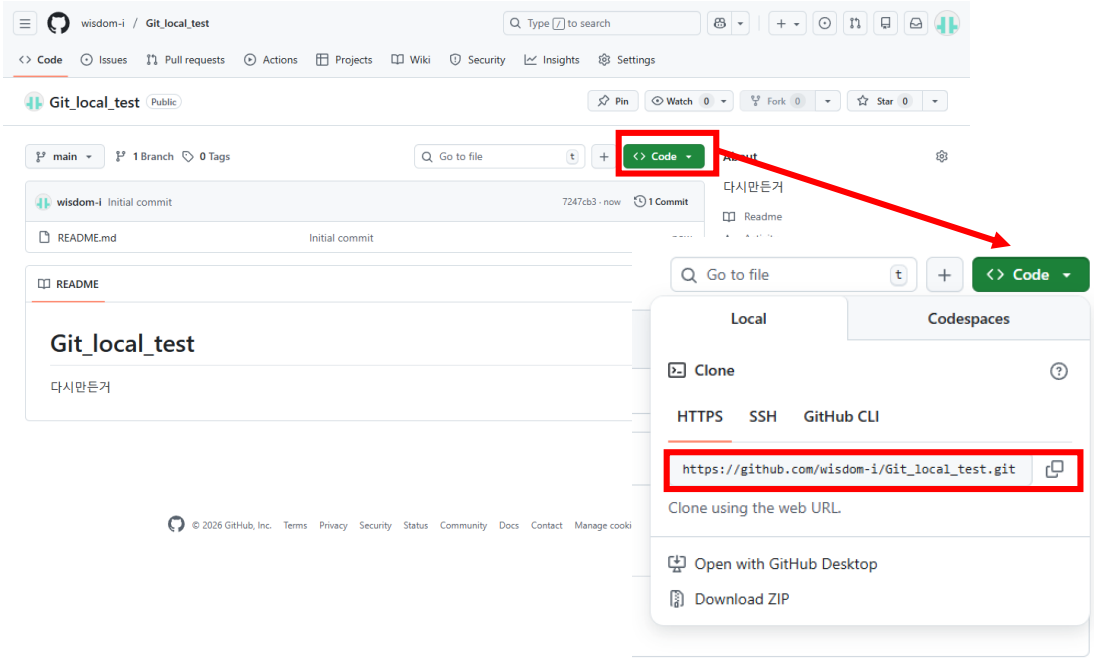
3. Local – repositories : 확인



Github 사용법 : Repositories to local

4. Repositories – local

- 새 프로젝트 생성

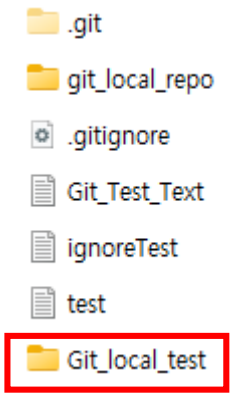


- 복사하고싶은 브랜치(폴더)에서 **$git clone 주소** 입력

```
JH@DESKTOP-BUR3LGT MINGW64 /c/Users/JH/Desktop/Git_Test (master)
$ git clone https://github.com/wisdom-i/Git_local_test.git
```

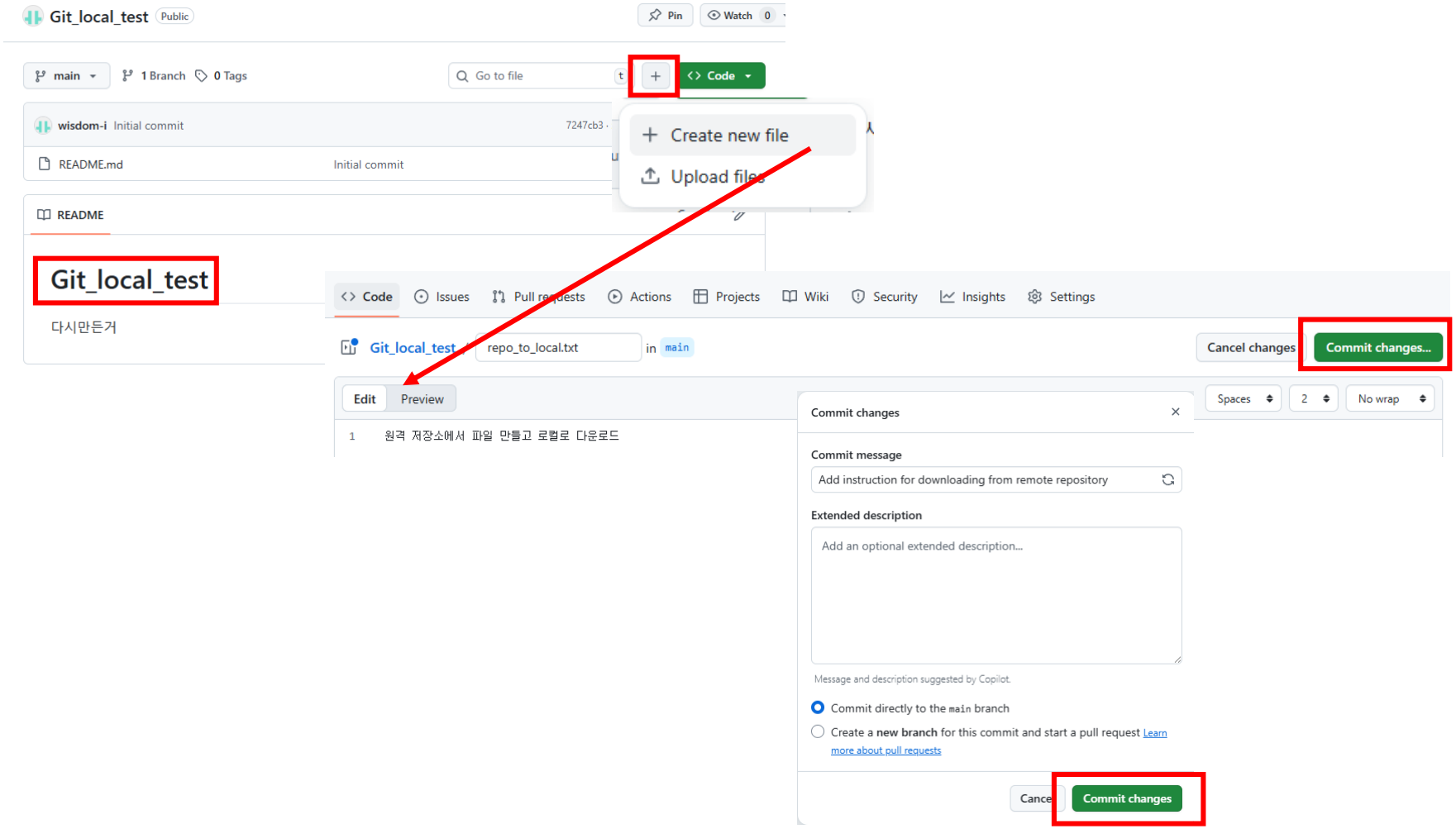
Github 사용법 : Repositories to local

5. Repositories – local : 확인



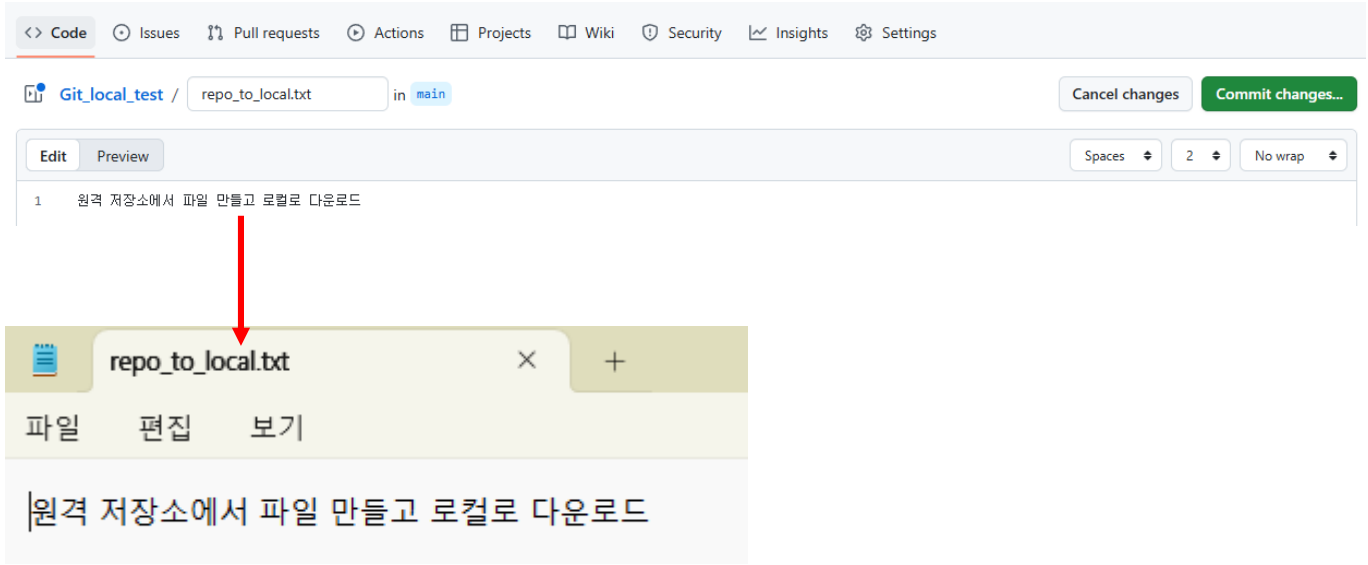
Github 사용법 : Repositories to local

5. Repositories – local : 확인

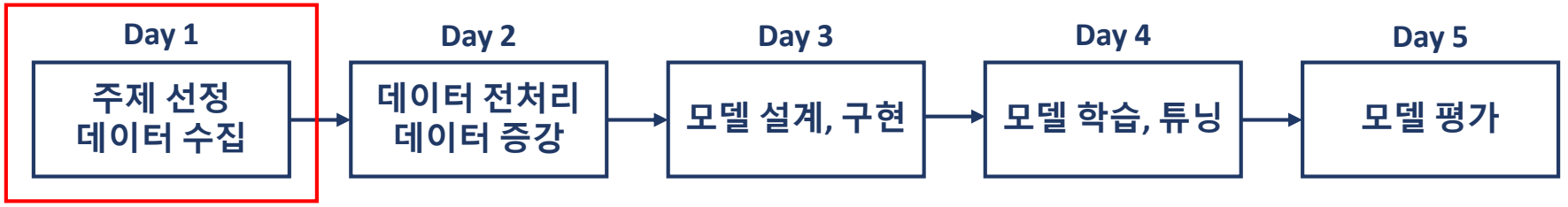


Github 사용법 : Repositories to local

5. Repositories – local : 확인



01/19 To-do



1. 주제 선정 및 데이터 수집

- 데이터 선정

  - 어떤 데이터를 분류해보고 싶은지

  - 예시 : 개 vs 고양이, 비행기 vs 자동차

?

- 데이터 수집

  - 수집 기준 : 라벨 당 200장 이상, 최소 2라벨 수집

  - 방법 : 직접 촬영 혹은 구글 서칭

Thank you