* **Git**
* 형상 기억 관리 프로그램
* 버전관리 프로그램
* 자신의 로컬 컴퓨터에 데이터를 저장하는 저장공간

1. git관리가 필요한 이유
   1. 개발 a to z 혼자 진행 불가
   2. 일반적으로 3~5인 팀으로 이루어져 개발 진행

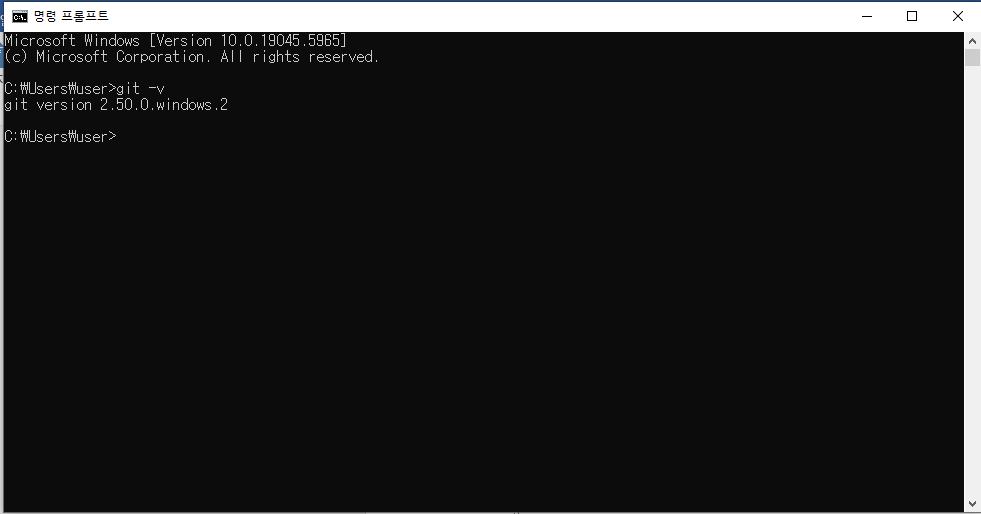
Ex1. 3명의 개발자가 각자 개발 후 파일 1명이 모든 파일 병합하는 경우, 충돌 발생

* 가 : 메인 페이지
* 나 : 로그인 페이지
* 다 : 회원가입 페이지

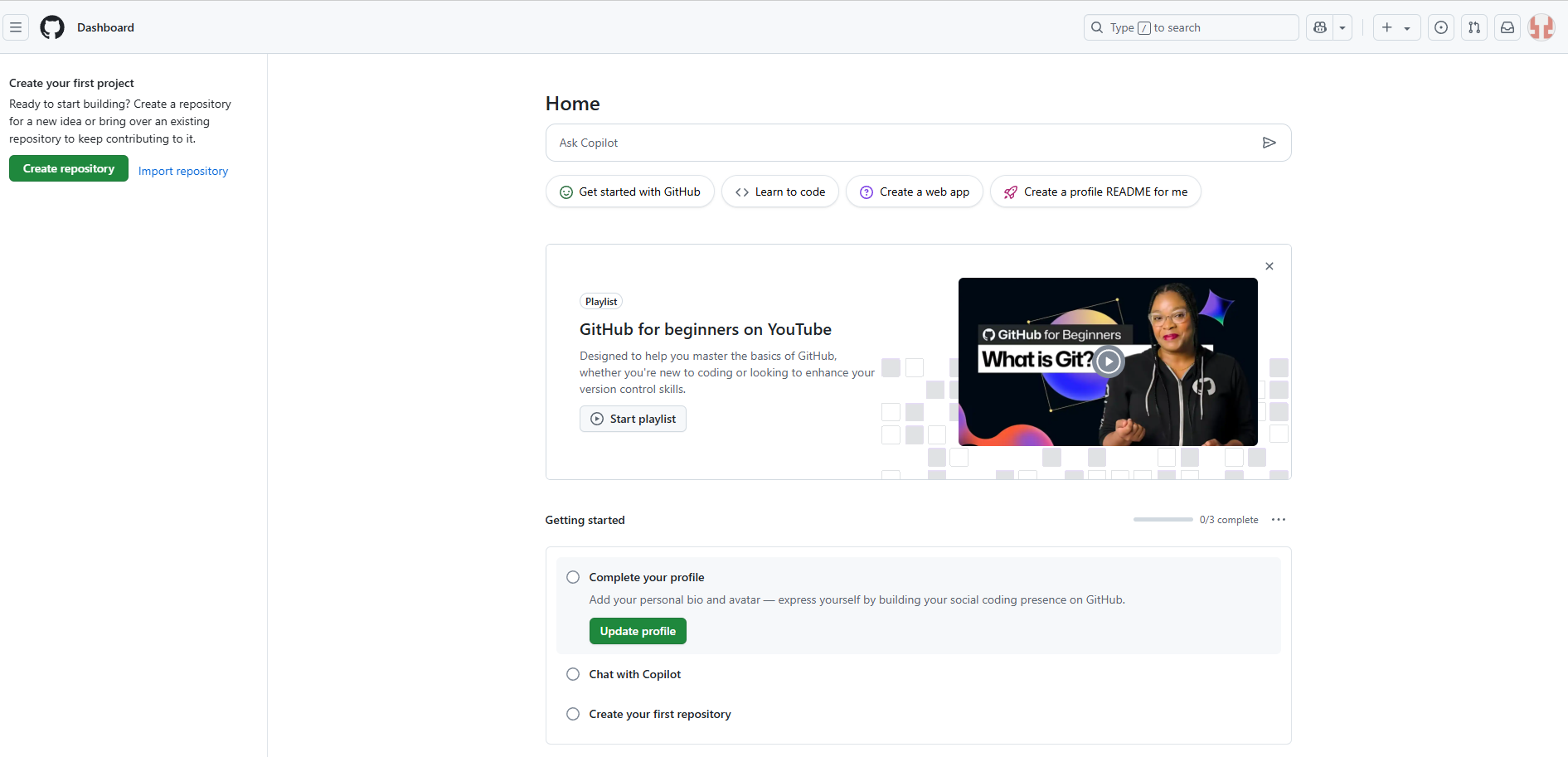
Ex2. 1인 혼자서 작업하다 후 참여 개발자에게 작업물 공유하여 +@ 내용이 생성시 병합과정 어려움

Ex3. 3인 작업시 (프리랜서), 정해놓은 작업물 공동의 폴더 업로드시 병합순서 정하기 어려움

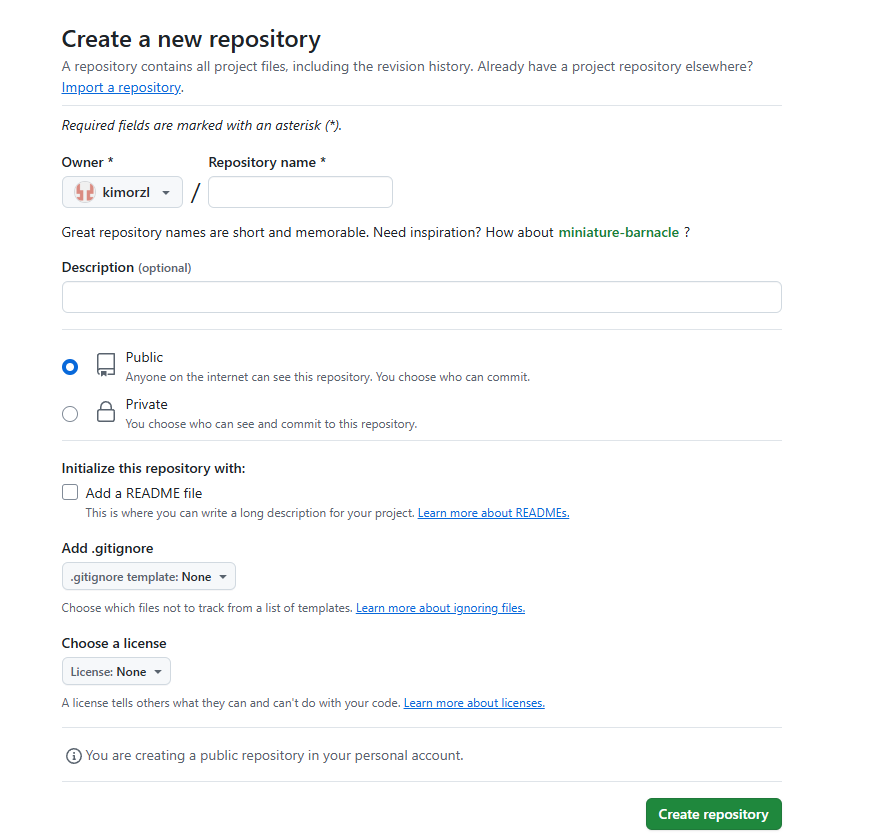
1. Git 사용법



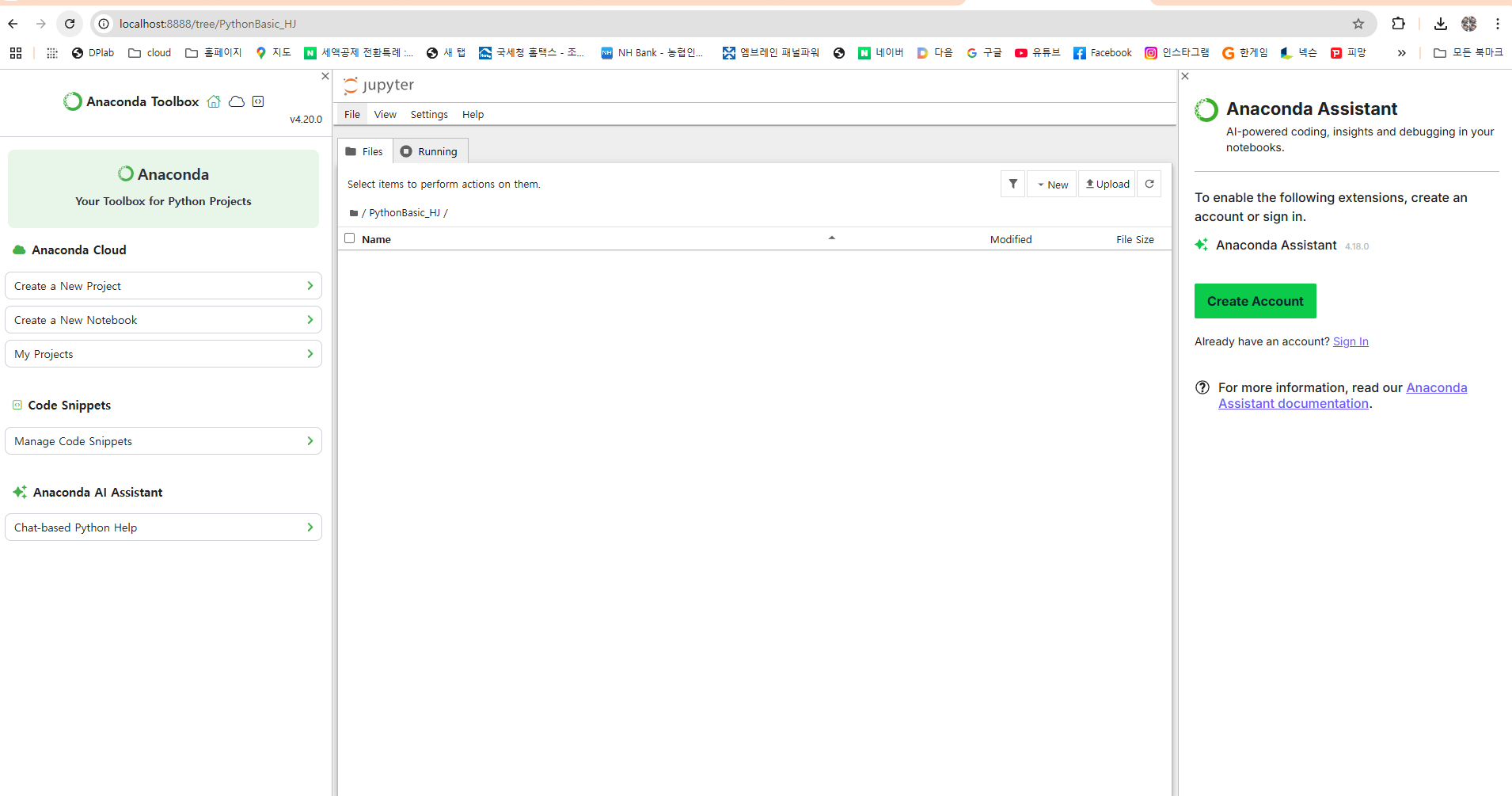
* 로컬 컴퓨터에 gitbash 관리자모드 설치 (cmd에서 git –v로 설치 확인)
* (mac 설치 시, 터미널에서 homebrew설치 후 mac 버전 다운로드)
* 설치 여부 확인 : 터미널 (cmd+space+git –v, 버전확인)



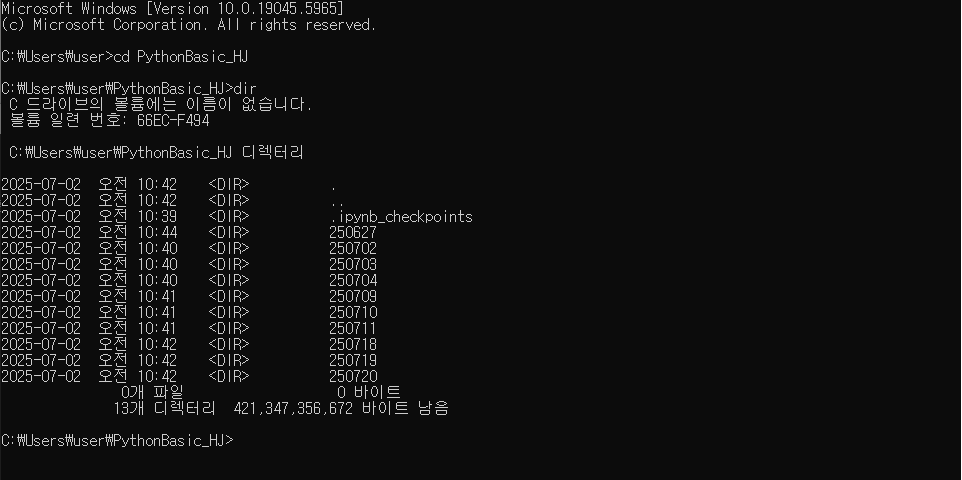
* Github.com에서 github 다운로드/계정생성



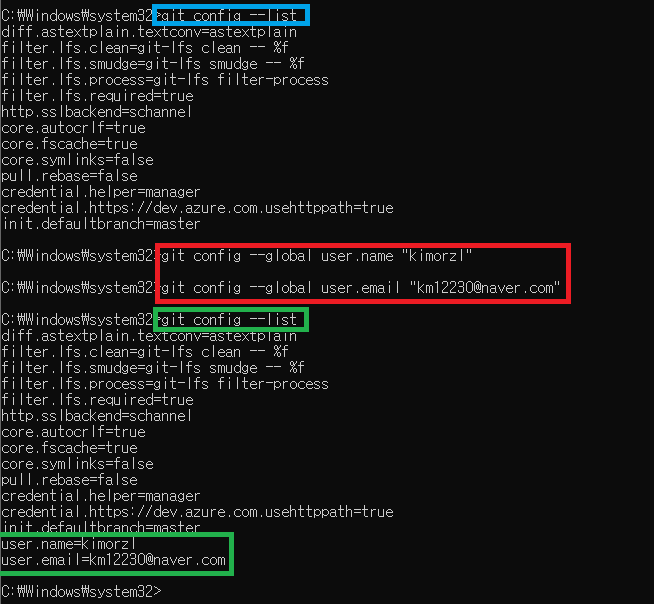
* Github repository 생성



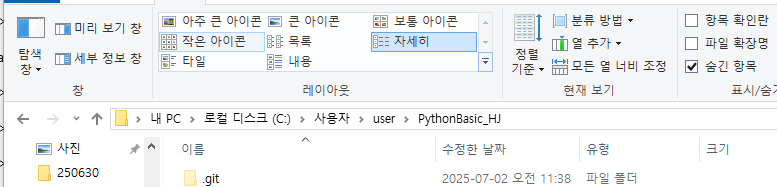
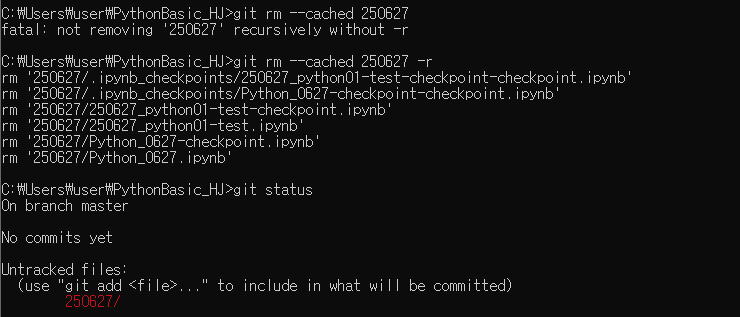
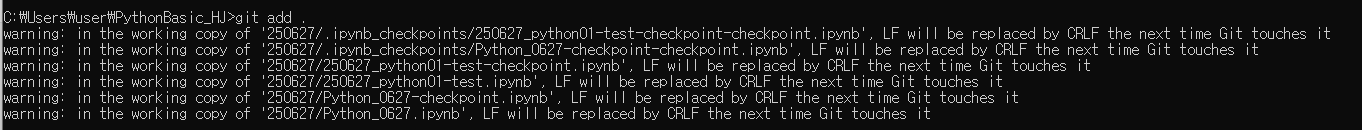
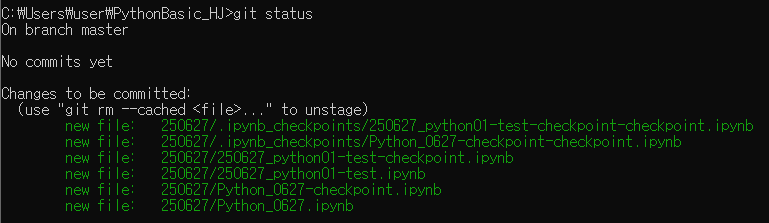
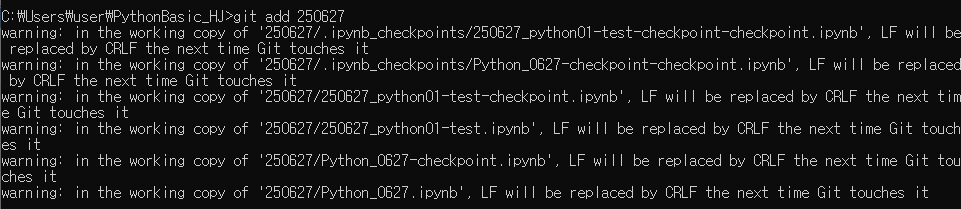
* Jupyter notebook 파일 경로 찾기
* 경로 mac, 이동 – 라이브러리 – admin – PythonBasic\_HJ > 날짜별 폴더
* 위치 파악, cmd에 ls입력하여 확인



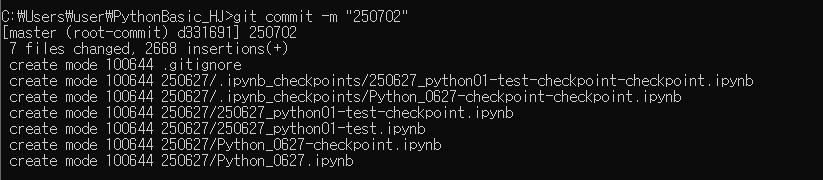
* Window의 경우 cmd에 cd>파일경로>dir입력



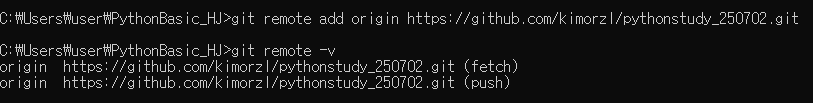
* 로컬 컴퓨터 > git 소유주 정보 입력 필요
* 관리자모드 cmd > git config --list 입력
* User.email, user.name 없을 경우 정보 삽입
* 관리자모드 cmd > git config --global user.name “입력하고 싶은 username”
* 관리자모드 cmd > git config --global user.email “입력하고 싶은 useremail”
* 관리자모드 cmd > git config --list하여 제대로 정보 입력되었는지 확인

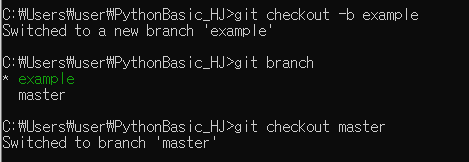
* Stageing Area(.git) 생성
* git init : init이라는 임시 저장공간을 만들어주는 명령어
* git add 250627 : 250627에 있는 파일을 .git으로 옮겨라
* git status : git의 현재 상태를 알려줘
* git add. : add. : 경로내 모든 파일을 업로드하겠다
  + 새로운 파일이 업데이트가 되어도 중복 업데이트가 되진 않음
  + 공폴더의 경우, 파일로 취급을 하지 않아 업데이트 되지 않음
* git rm –cached (파일명) -r : 캐시 되어지 (파일명)을 git에서 지우겠다
  + git을 웹하드로 사용하면 안되는 이유
    - 무료 버전 사용시, cach값이 100메가가 넘어가는 순간 초기화 작업필요



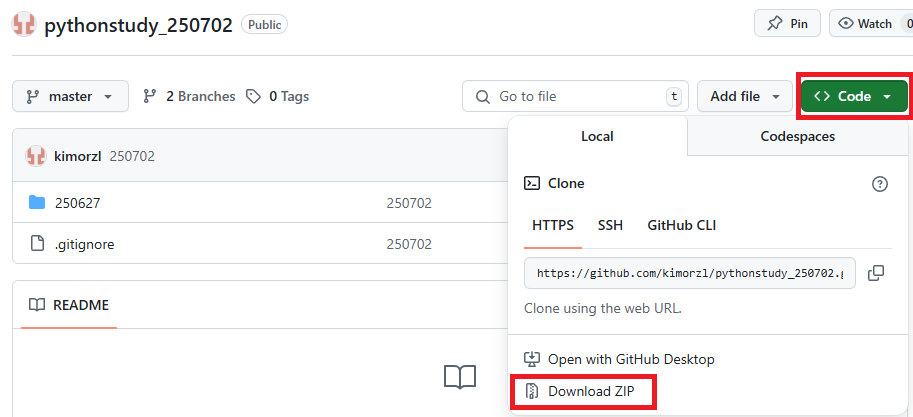
* git hub에 commit 이름 부여
* gi tcommit –m ”커밋명”
  + 동일한 이름 부여는 신중해야함 (파일 관리 어려움)
  + git에 commit하는 순간 기존 git서버에는 더 이상 파일이 남아있지 않음



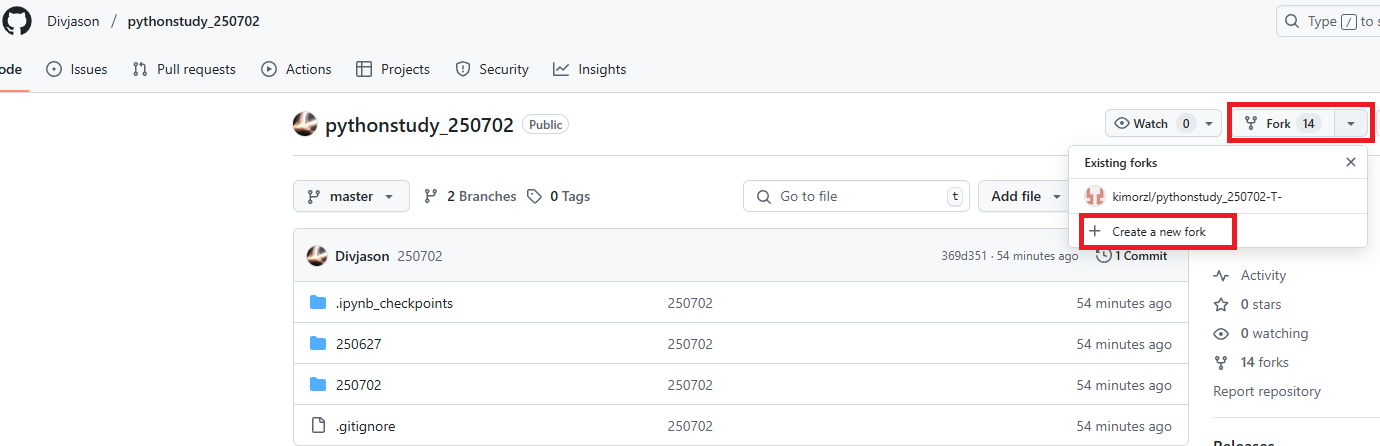
* git remote add origin <https://github.com/kimorzl/pythonstudy_250702.git>
  + fetch : 데이터를 받을 수 있는 상태 준비가 되었다
  + push : 명력어 및 데이터를 올릴 (입력받을) 준비가 되었다.
  + Branch : repository내 작업이 가능한 각각의 파일



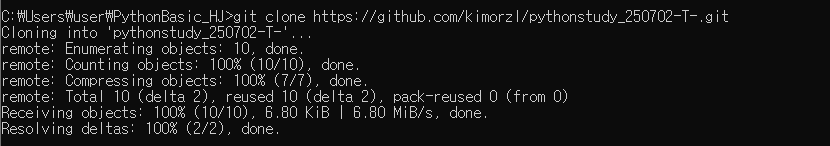
* Branch 신규생성
  + git checkout –b 브랜치명
  + git branch : 내가 있는 브랜치 위치를 알려줘
  + git checkout 브랜치명 : 브랜치명으로 이동해줘
* fetch 방법



* + code – download zip (확실하나, 단무지 스타일) : git에 있는 모든 데이터를 다운



* + fork : 타인의 repository를 가져오는 행위
    - 선생님 repository 주소 (<https://github.com/Divjason/pythonstudy_250702>)



* + Git clone repository 주소 : 복사해오고 싶은 repository 주소를 입력
  + git pull repository 주소 : 이미가져온 것 이외의 가져오지 못한 추가 정보 다운

# 객체 지향 언어

* 어떤 자료를 생성시 각각의 객체를 생성하는 것

# 절차지향

* 그때마다 필요한 객체를 각각 선언하여 사용

# 프로토타입

* 창시자가 class, list 등 다양한 설계도면 버전으로 구성된 언어
* Class는 어떠한 문법 체계