

ORIENTATION

▶ 간단한 강사 소개



강보람

010-5511-9220



(디지털컨버전스)

공공데이터 융합

자바/스프링

개발자 양성과정



2022.10.12 ~ 2023.03.27

900시간

약 113일

(5.5개월 과정)

▶ 수업 관련

✓ 커리큘럼

능력단위	수업내용
프로그래밍 언어 활용 (40시간, 5일)	JAVA
프로그래밍 언어 응용 (56시간, 7일)	
공공데이터 수집을 위한 객체지향 프로그램 구현 (88시간, 11일)	
SQL 활용 (32시간, 4일)	Oracle
데이터베이스 구현 (40시간, 5일)	
SQL 응용 (32시간, 4일)	JDBC

▶ 수업 관련

✓ 커리큘럼

능력단위	수업내용
화면설계 (56시간, 7일)	HTML, CSS, JavaScript, jQuery
UI 구현 (80시간, 10일)	
요구사항 확인 (56시간, 7일)	Servlet/JSP, AJAX
서버프로그램 구현 (88시간, 11일)	
애플리케이션 설계 (56시간, 7일)	SEMI Project
애플리케이션 배포 (40시간, 5일)	

▶ 수업 관련

✓ 커리큘럼

능력단위	수업내용
인터페이스 구현 (56시간, 7일)	Framework (MyBatis, Spring)
통합 구현 (56시간, 7일)	
공공데이터 시스템 프로젝트 (118시간, 15일)	FINAL Project

▶ 수업 관련

✓ 전달 사항

능력단위평가

꾸준한 복습

출석 100%

수업시간 / 점심시간

▶ 수업 관련

✓ 전달 사항

능력단위평가

꾸준한 복습

출석 100%

수업시간 / 점심시간

▶ 수업 관련

✓ 전달 사항

능력단위평가

꾸준한 복습

출석 100%

수업시간 / 점심시간

▶ 수업 관련

✓ 전달 사항

능력단위평가

꾸준한 복습

출석 100%

수업시간 / 점심시간

▶ 수업 관련

✓ 전달 사항

능력단위평가

꾸준한 복습

출석 100%

수업시간 / 점심시간

09:00 ~ 09:50

10:00 ~ 10:50

11:00 ~ 11:50

12:00 ~ 12:50

12:50 ~ 14:00

점심시간

14:00 ~ 14:50

15:00 ~ 15:50

16:00 ~ 16:50

17:00 ~ 17:50

▶ 자기소개



**실수를 두려워하지 마세요.
질문을 무서워하지 마세요.**

GitHub를 이용한 형상관리

▶ GitHub를 이용한 형상관리

✓ 형상관리란?

소프트웨어의 변경사항을 체계적으로 추적하고 통제하는것

즉, 이를 통해 어떤 문서나 파일이 변경됐을 경우 변경 원인과 변경사항을 확인할 수 있음

* 형상관리 도구 : CVS, SVN, Git 등등..

✓ GitHub

클라우드 서버를 사용해서 로컬에서 버전 관리한 소스코드를 업로드하여 공유할 수 있음

(여러 사람이 서로 간에 소스코드를 공유해가며 개발 할 수 있음 == 협업 가능)

* <https://github.com/>

▶ GitHub관련 주요 용어

✓ 원격저장소 (Remote Repository)

여러 사람이 함께 공유해서 사용하는 저장소 (GitHub 서버 또는 클라우드에 존재하는 저장소)

✓ 로컬저장소 (Local Repository)

로컬저장소는 내 PC에 저장되는 개인용 저장소 (자신의 컴퓨터에 있는 저장소)

* 기본적으로 로컬저장소(내 PC)에서 작업을 수행하고 그 파일을 원격저장소(Github상)에 업로드하는 방식

▶ GitHub관련 주요 용어

✓ clone

GitHub상에 만들어져 있는 **원격저장소(Remote Repository)**를 로컬(내 PC)로 **복제**할 때 사용

✓ commit

소스코드의 업데이트를 **확정**하고자 할 때 사용
(단지 확정만 지었을 뿐 원격저장소에는 반영되지 않음)

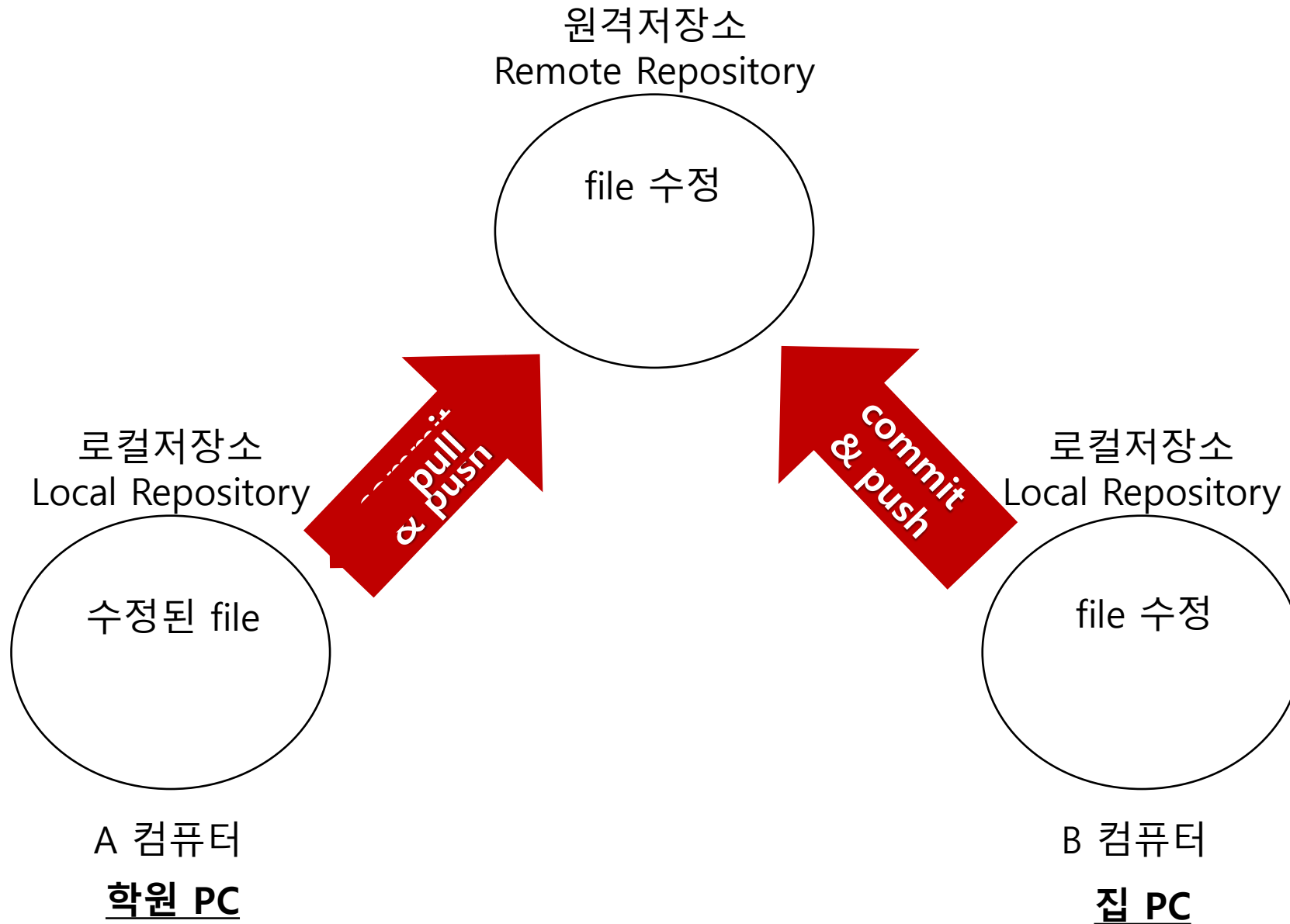
✓ push

commit한 내용을 원격저장소에 본격적으로 올릴 때 사용
(즉, 로컬저장소의 변경사항을 원격저장소로 **업로드**)

✓ pull

원격저장소의 변경사항을 로컬저장소에 반영시키는 것
(즉, 원격저장소의 변경사항을 로컬저장소로 **다운로드**)

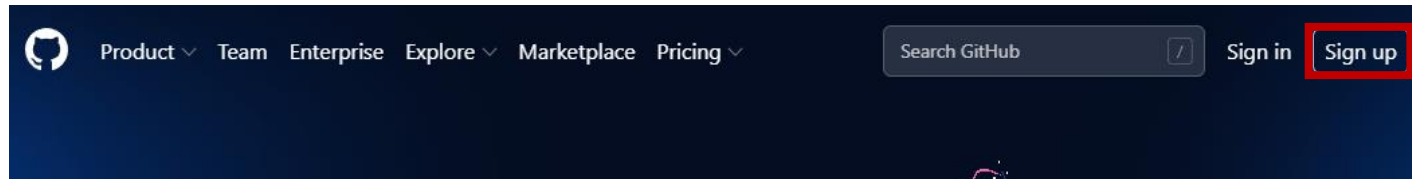
▶ GitHub관련 주요 용어



▶ GitHub 준비사항

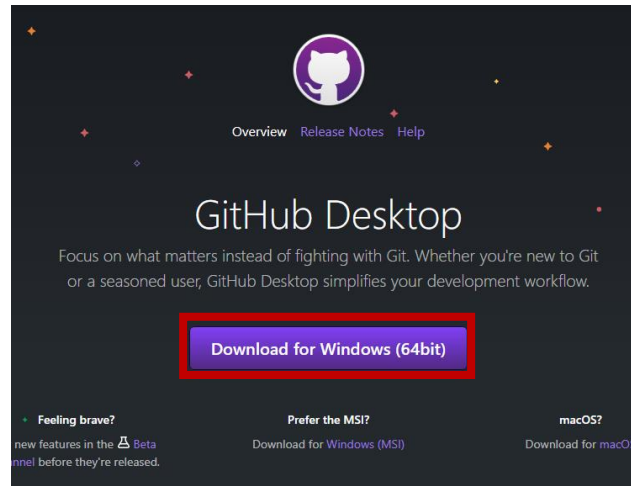
✓ GitHub 사이트 회원가입

<https://github.com/> 접속해서 회원가입



✓ GitHub Desktop 프로그램 설치 (GUI 툴)

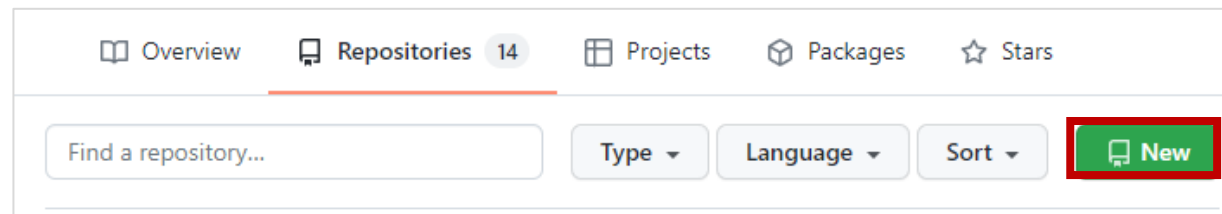
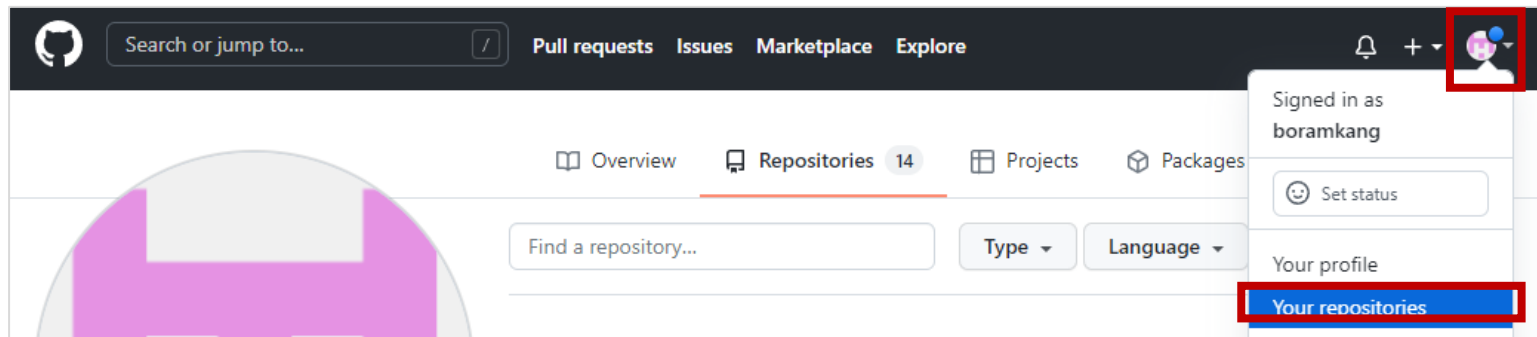
<https://desktop.github.com/> 접속해서 해당 프로그램 설치하기



▶ GitHub 이용하기

✓ 원격저장소 만들기

<https://github.com/> 접속하여 로그인 후 Your repositories > NEW 클릭



▶ GitHub 이용하기


✓ 원격저장소 만들기

Create a new repository

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

Owner *

Repository name *

 boramkang ▾

/

xxx(레파지토리명) ✓


Great repository names are

Your new repository will be created as xxx-.


How about [legendary-eureka?](#)

Description (optional)

xxxxx(해당 레파지토리설명)

☐  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](#).

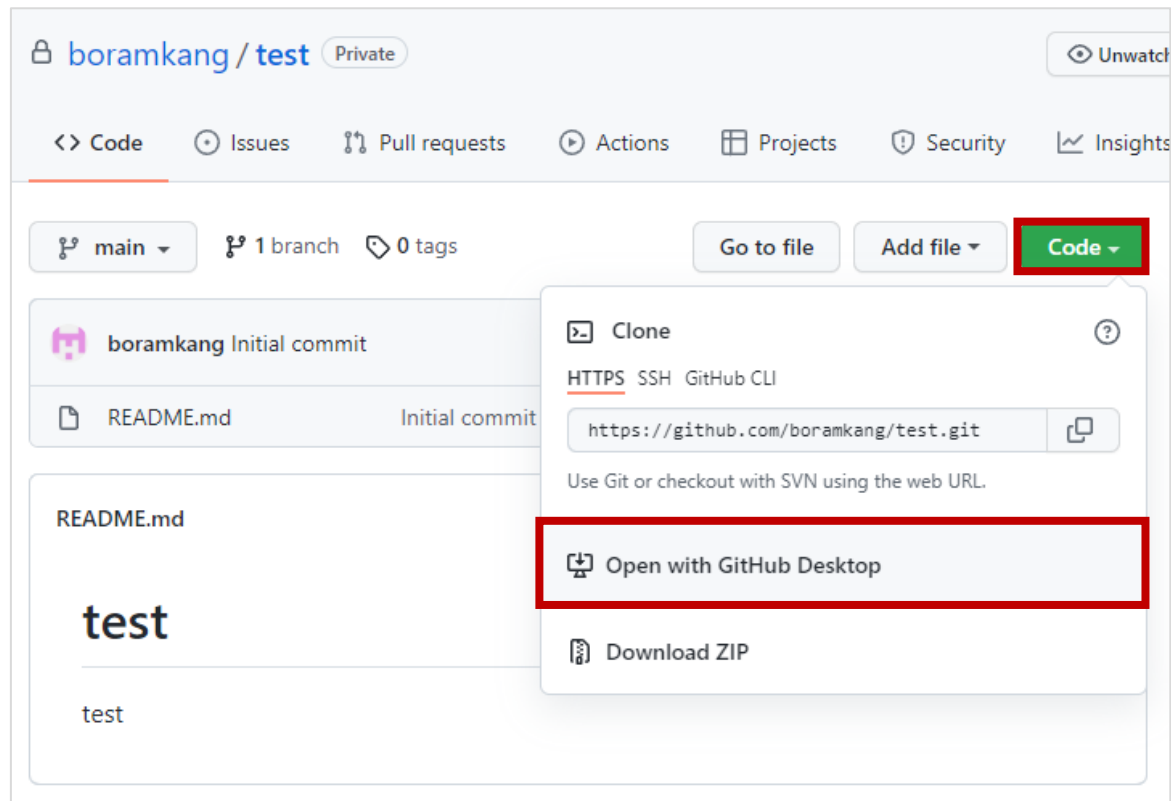
 You are creating a private repository in your personal account.

Create repository

▶ GitHub 이용하기

✓ 원격저장소를 내 PC로 clone하기 (로컬저장소만들기)

만들어진 원격저장소에 들어가 Code > Open with GitHub Desktop 클릭

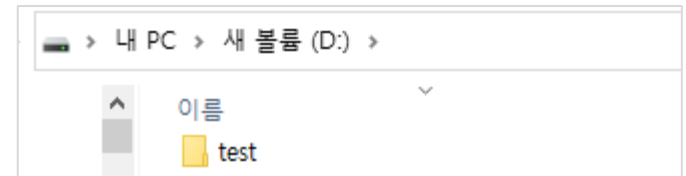
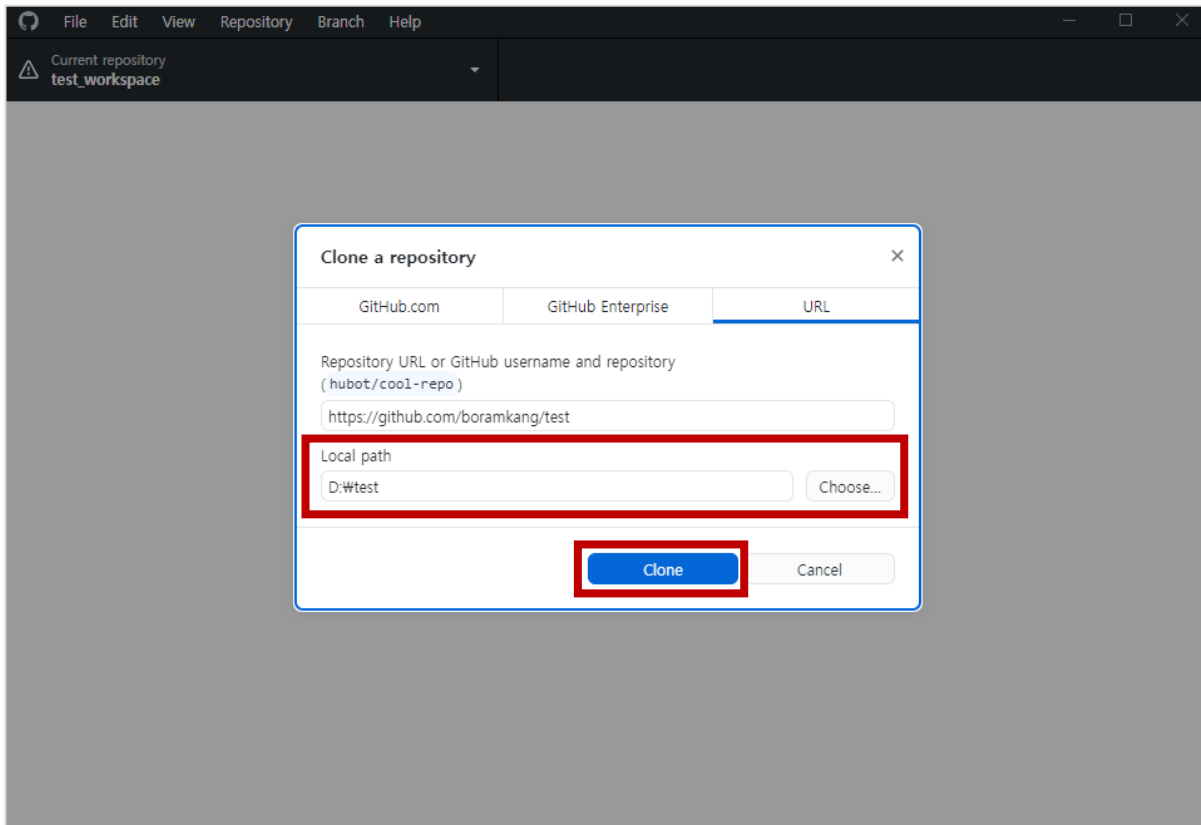


▶ GitHub 이용하기

✓ 원격저장소를 내 PC로 clone하기 (로컬저장소만들기)

내 PC의 어떤 경로에 저장 시킬꺼지(즉, clone할꺼지)

Local path 수정 후 clone 버튼 클릭 => 해당 경로에 폴더가 만들어진 걸 확인할 수 있음



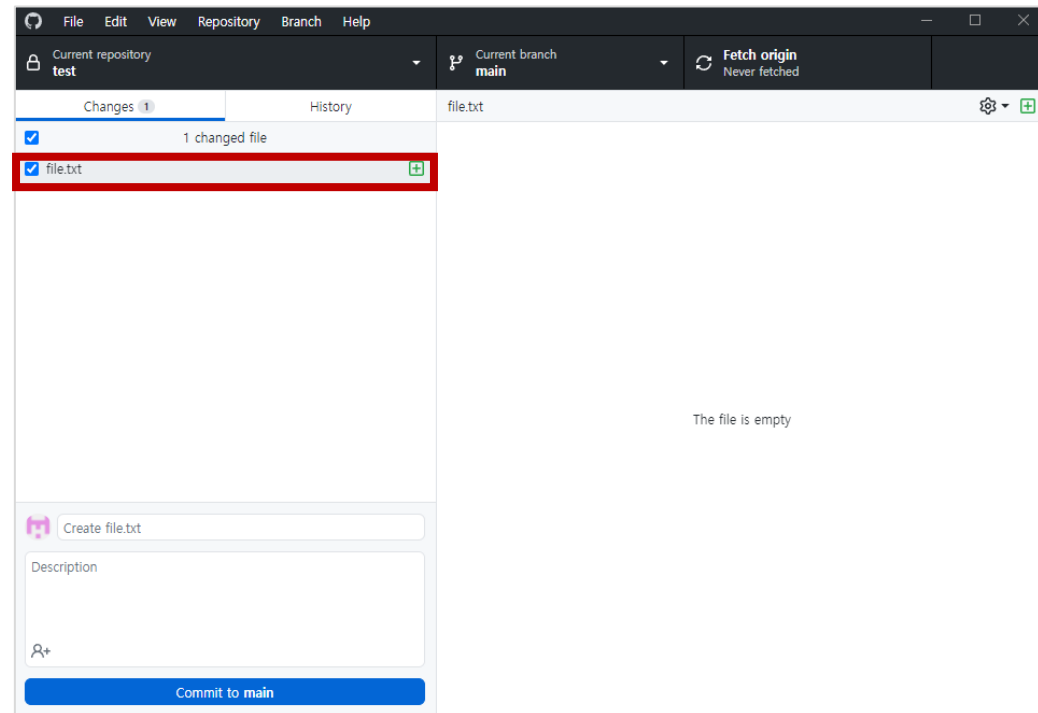
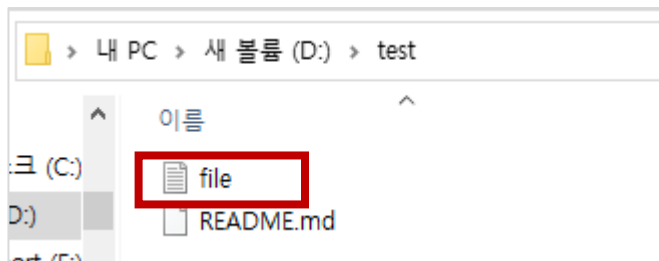
▶ GitHub 이용하기

✓ 로컬저장소의 변경사항을 원격저장소에 업로드(commit & push)

로컬저장소에 간단하게 파일을 하나 만든 후

GitHub Desktop을 다시 보면 Changes 탭에 해당 파일이 보여짐

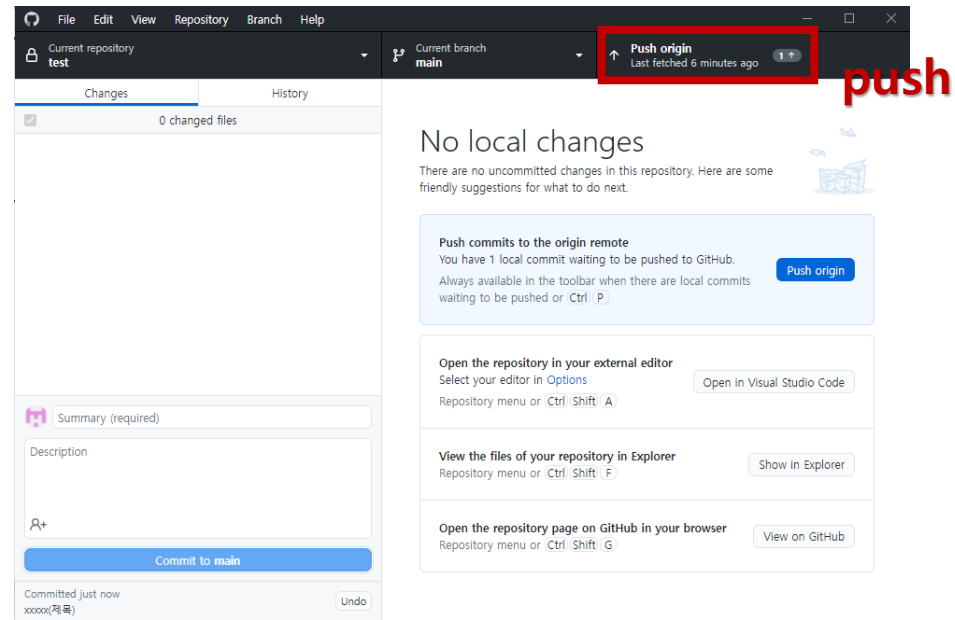
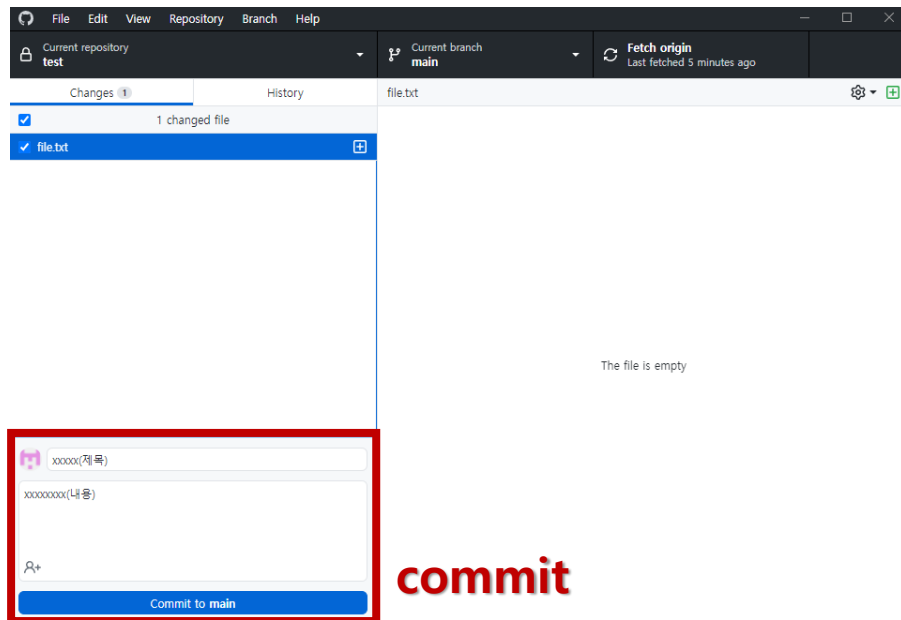
* Changes 탭 : 원격저장소와 로컬저장소를 비교하여 로컬저장소에 변경사항이 생겼을 경우
이를 원격저장소에 올릴 것인지 해당 변경사항의 파일이 보여짐



▶ GitHub 이용하기

✓ 로컬저장소의 변경사항을 원격저장소에 업로드(commit & push)

해당 변경사항의 파일을 원격저장소로 업로드 시키고자 한다면 commit 후 push 진행
이때 commit message를 작성하는데
제목과 내용을 작성하여 "어떤 파일"을 "어떻게" "왜" 변경했는지 상세 작성 후
commit and push 진행함



▶ GitHub 이용하기

✓ 원격저장소의 변경사항을 로컬저장소로 다운로드 (pull)

혹시라도 원격저장소에 변경사항이 생겼을 경우
이를 로컬저장소로 다운로드 받고자 한다면 pull을 하면 됨
(pull 내역이 바로 보여지지 않는다면 Fetch origin 버튼(새로고침 같은거)을 클릭하면 됨)

