

# 오픈소스SW기초 (2024-2)

## 3. GitHub 기초 기능 입문 II

국립금오공과대학교  
컴퓨터공학과 / 인공지능공학과  
김 경 수

# 목차

- 여러 저장소 다루기
- 저장소의 복제와 연동
- 종합 실습 I - GitHub Desktop 기초 기능 연습
- Appendix. GitHub Desktop의 주요 기능 요약

# 학습 목표

- ① GitHub Desktop을 이용하여 두 개 이상의 저장소를 생성하고 운용, 관리하는 방법을 이해하고 간단한 사례를 이용하여 실습할 수 있다.
- ② GitHub의 Clone과 Pull 기능을 이용하여 저장소를 복제하여 여러 기기에서 원격 운용하는 방법과 절차를 이해하고 전 과정을 숙달할 수 있다.
- ③ GitHub Desktop을 이용한 저장소의 생성과 파일의 변경 사항을 관리하는 일련의 전 과정을 숙달하여 논리적으로 설명할 수 있다.

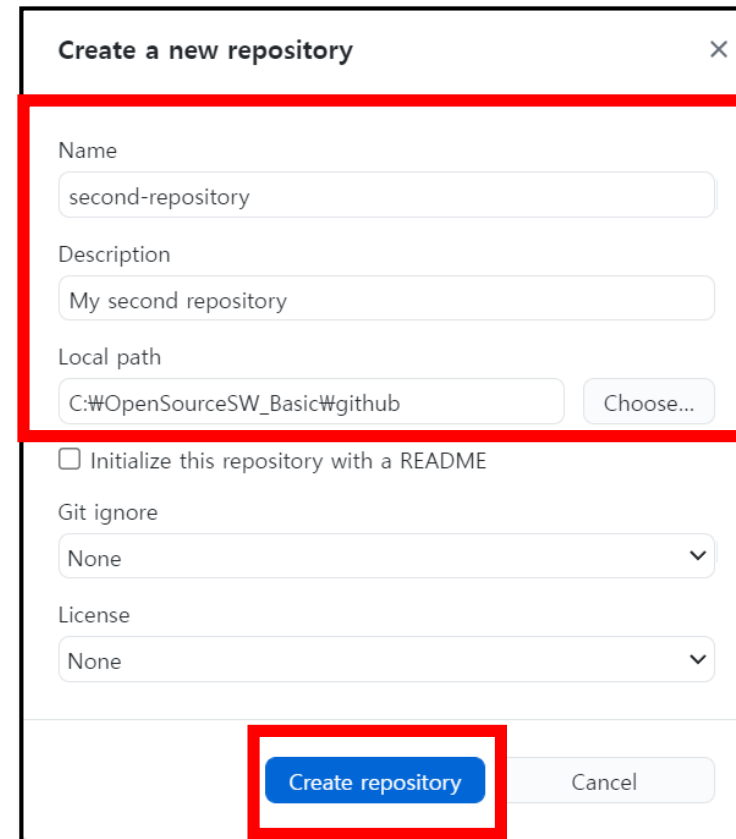
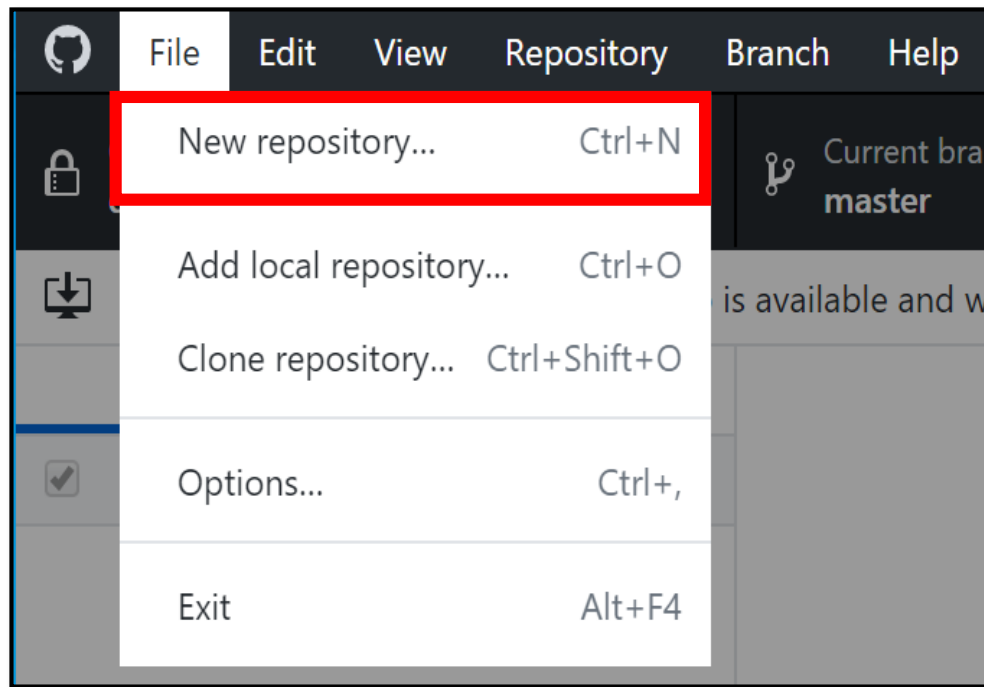
# 여러 저장소 다루기

# 여러 저장소 다루기

- Git에서는 주제 또는 목적에 따라 다수의 저장소를 생성하여 관리할 수 있음.
- 또한 하나의 원격 저장소에 대해 여러 컴퓨터 상의 로컬 저장소를 연동하여 사용할 수 있으며, 이를 통해 각 기간간의 문서 내용을 최신 상태로 유지하고 효과적으로 관리할 수 있음.
- GitHub Desktop에서는 여러 저장소를 다루기 위한 git 명령 중 저장소 생성 기능 외에도 remove, clone, pull과 같은 다양한 저장소 관리 기능을 제공함.

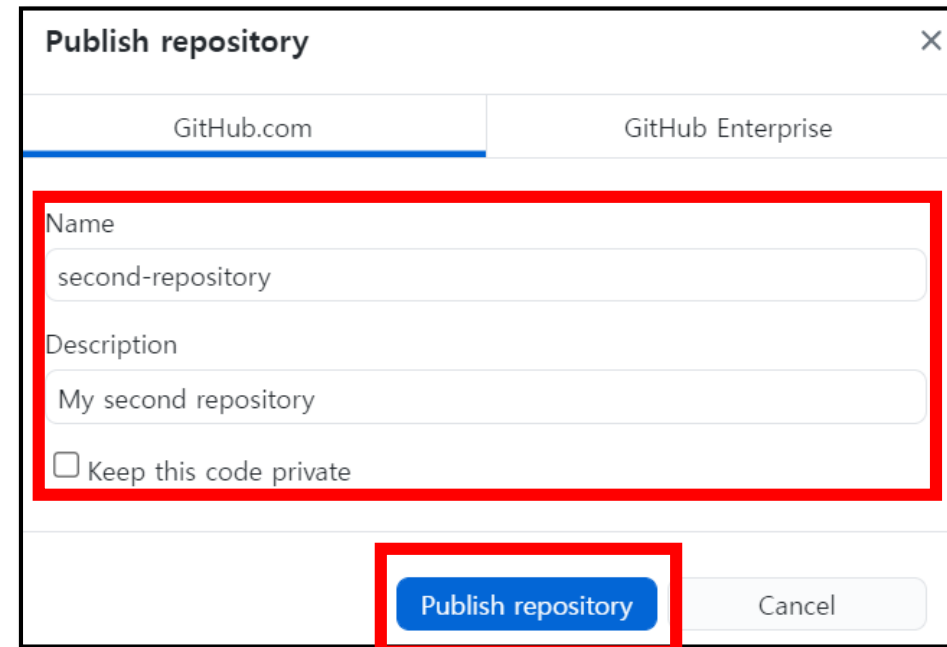
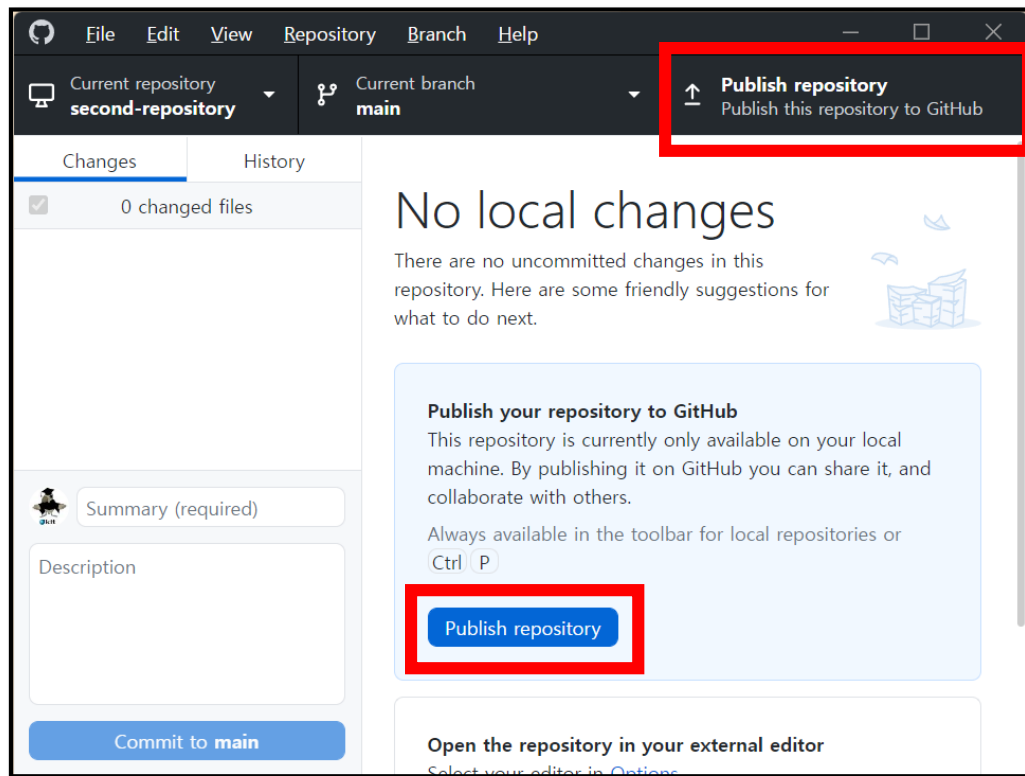
# 여러 저장소의 생성과 업로드

- “File – New repository” 클릭
- 새로운 저장소 이름과 설명을 아래 예제와 같이 입력한 후 “Create repository” 버튼 클릭

A screenshot of the 'Create a new repository' dialog box. The dialog has a title bar with a close button. It contains several input fields: 'Name' (filled with 'second-repository'), 'Description' (filled with 'My second repository'), and 'Local path' (filled with 'C:\OpenSourceSW\_Basic\github'). There is a 'Choose...' button next to the local path field. Below these fields are three checkboxes: 'Initialize this repository with a README' (unchecked), 'Git ignore' (set to 'None'), and 'License' (set to 'None'). At the bottom, there are two buttons: 'Create repository' (highlighted with a red box) and 'Cancel'.

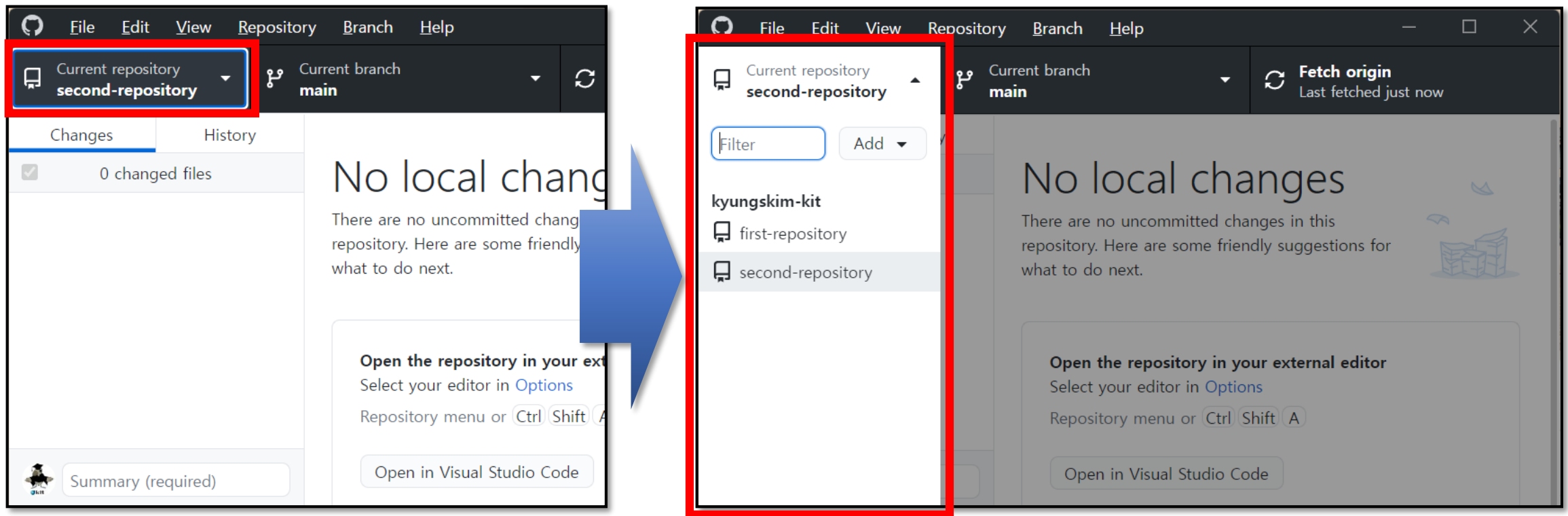
# 여러 저장소의 생성과 업로드

- “second-repository”가 생성되면 Publish repository 버튼을 클릭하여 생성된 로컬 저장소를 원격 저장소로 업로드



# 여러 저장소의 생성과 업로드

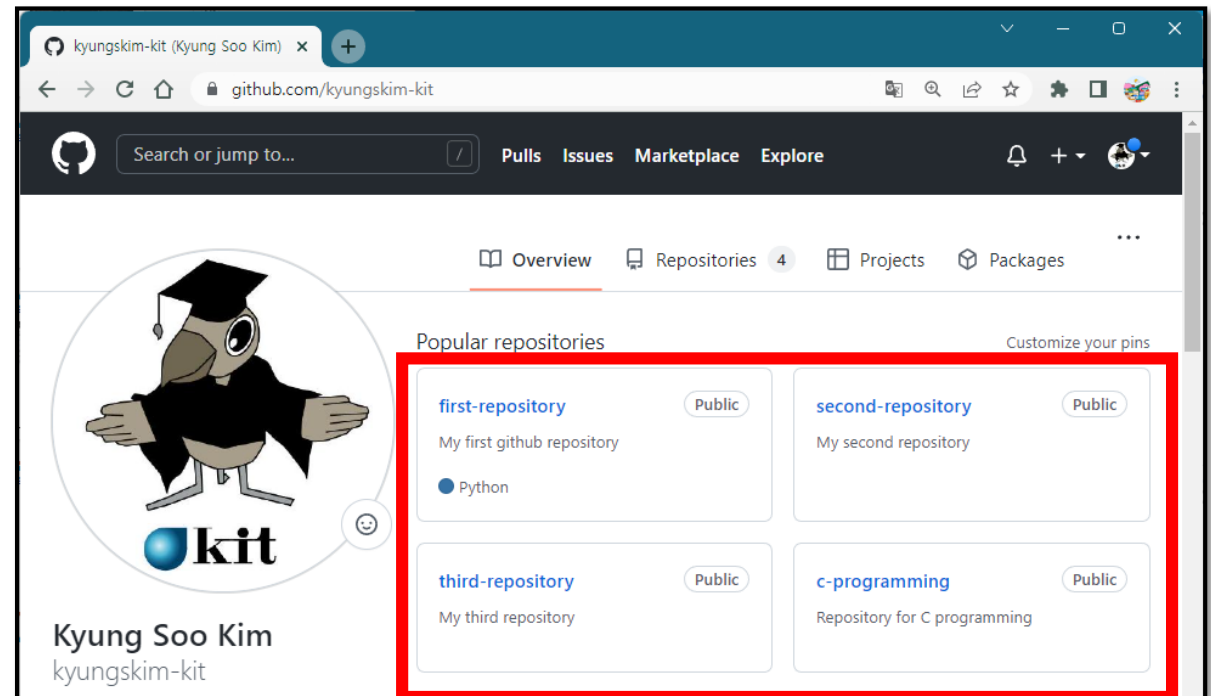
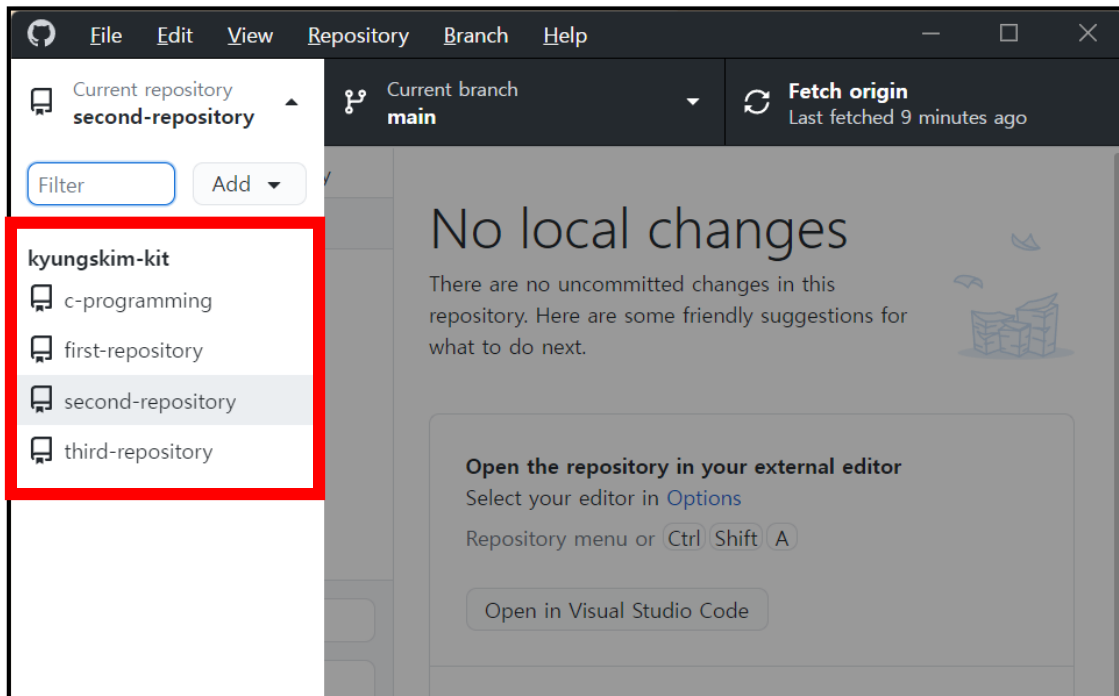
- GitHub Desktop 화면 왼쪽 상단의 “Current repository”를 클릭하면 현재 사용 가능한 저장소 목록을 확인할 수 있음.
- 해당 목록에 표시된 저장소를 클릭하여 현재 작업중인 저장소를 변경할 수 있음.





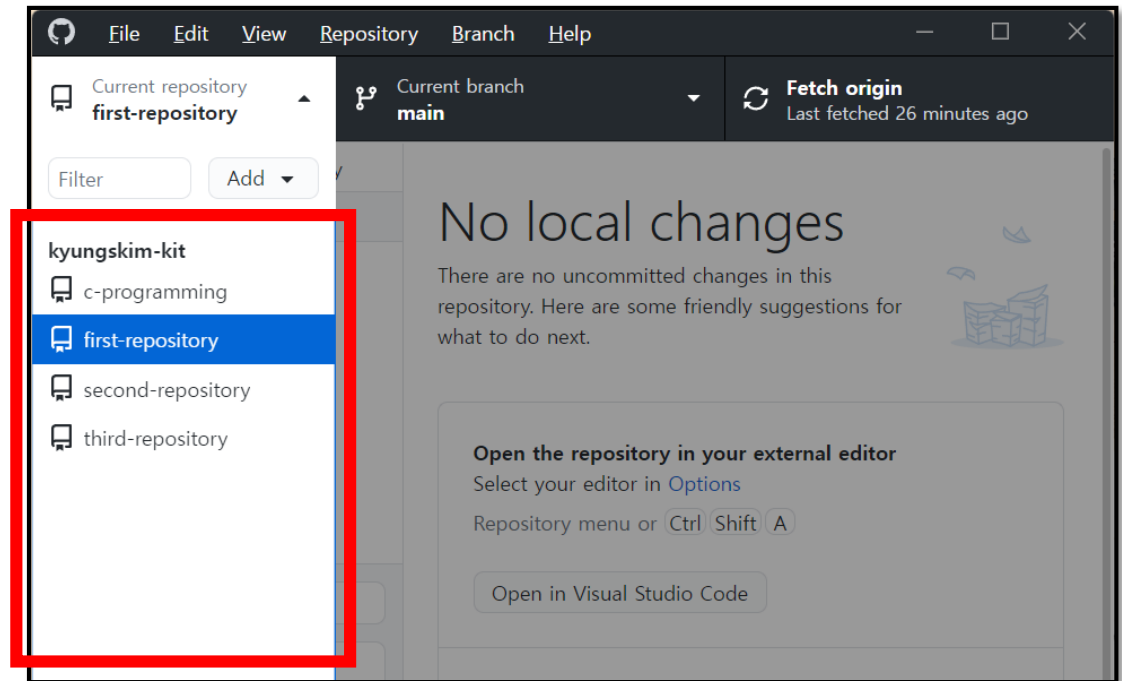
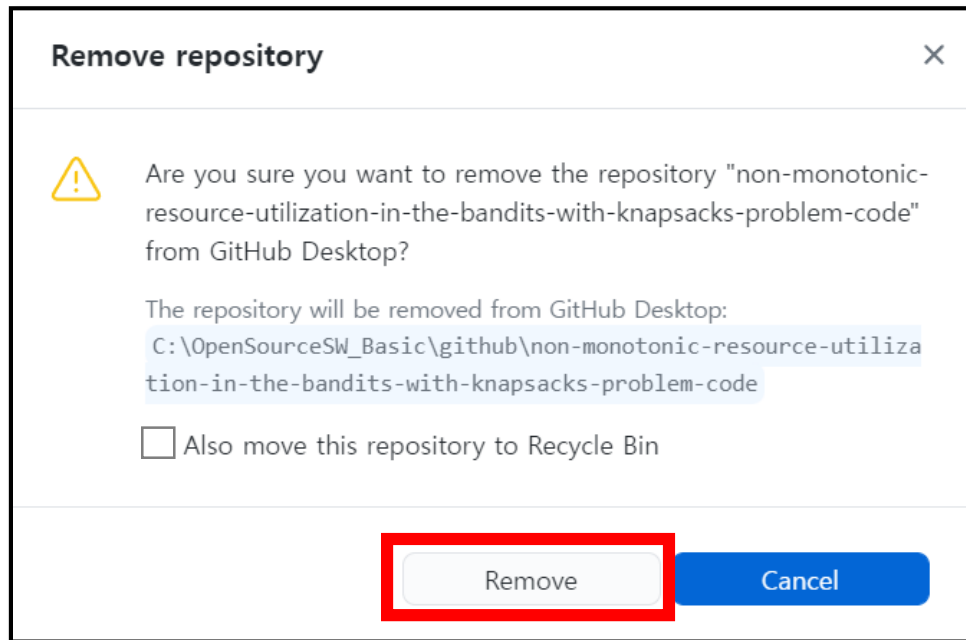
# 여러 저장소의 생성과 업로드

- 앞서 설명한 원리로 아래의 이름을 갖는 로컬/원격 저장소 두 개를 추가로 생성
  - third-repository
  - c-programming



# 저장소 삭제(remove)

- 화면 왼쪽 상단의 “Current repository”를 클릭한 후 삭제하기를 원하는 저장소를 선택
- 화면 상단의 메뉴바에서 “Repository-Remove”를 클릭한 후 “Remove repository” 창이 나타나면 “Remove”버튼을 클릭
  - 본 실습에서는 방금 생성한 “third-repository” 삭제해본다.



# 여러 저장소의 복제와 연동

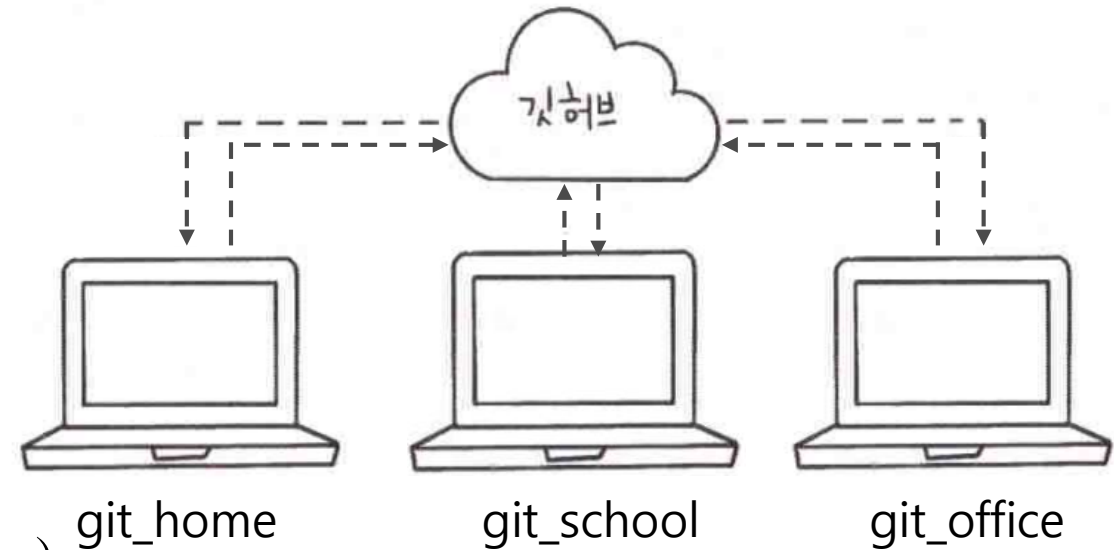
# 여러 장비에서 git을 활용하는 예

- 내가 가지고 있는 GitHub 계정을 활용하여 여러 컴퓨터에서 작업하고 싶다면 어떻게 해야 하는가?

## • 문제 상황

- 나의 원격 저장소는 1개
- 나의 로컬 저장소는 여러 개

(→ 집 컴퓨터, 사무실 컴퓨터, 학교 컴퓨터, ...)



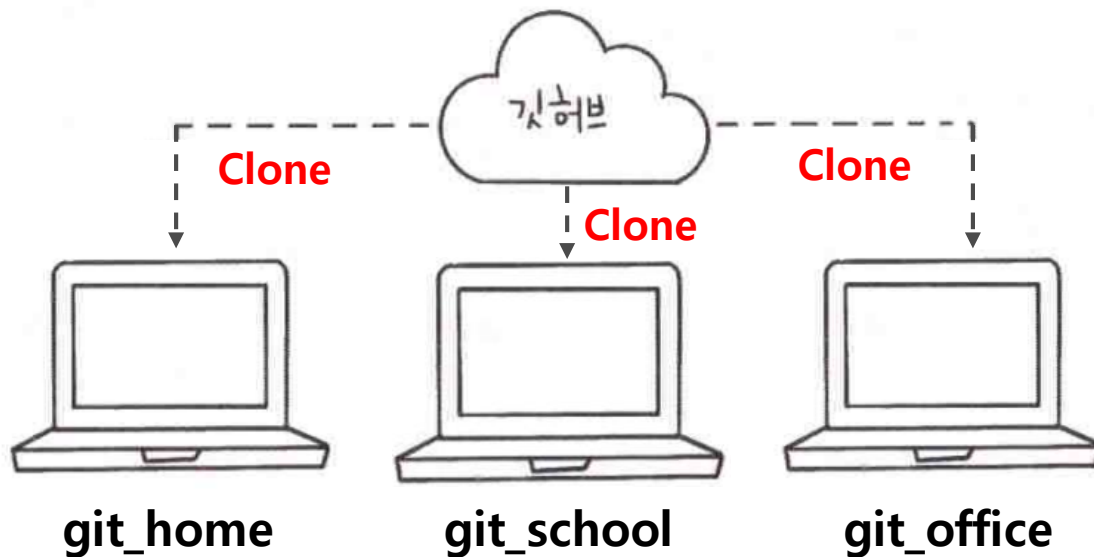
- [문제] 집에서 작업한 결과를 원격 저장소에 반영하면, 그 결과가 사무실 컴퓨터와 학교 컴퓨터의 로컬 저장소에도 자동으로 반영되는가?

# 저장소의 복제와 연동 기능

- GitHub Desktop에서 제공하는 저장소 복제 및 연동 기능
  - Clone repository
  - Pull repository

# Clone repository

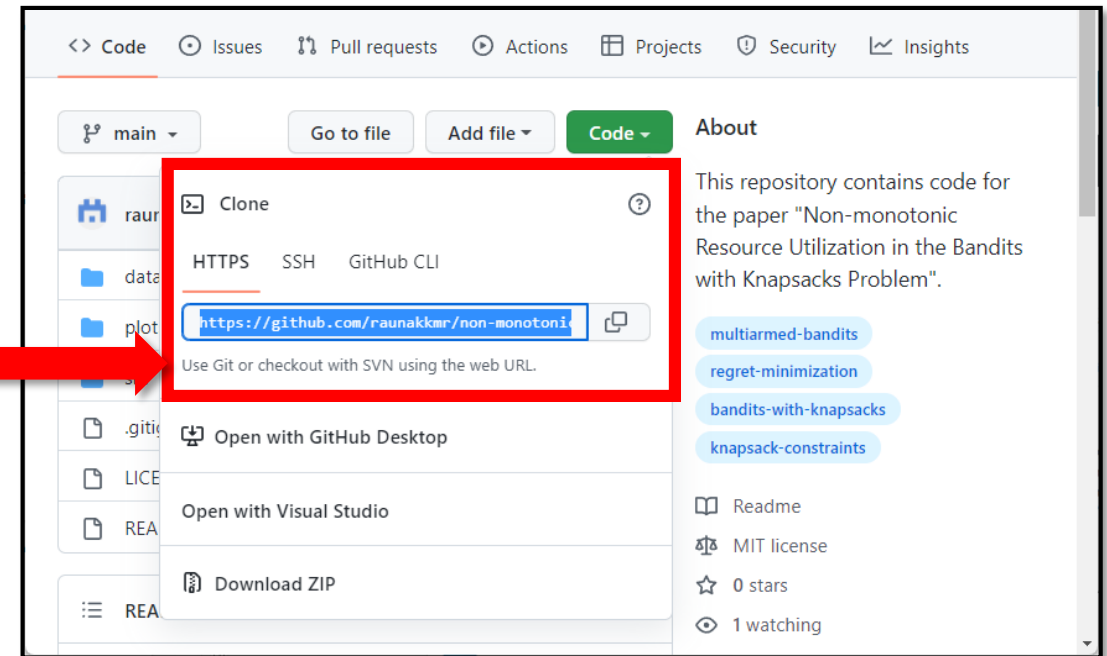
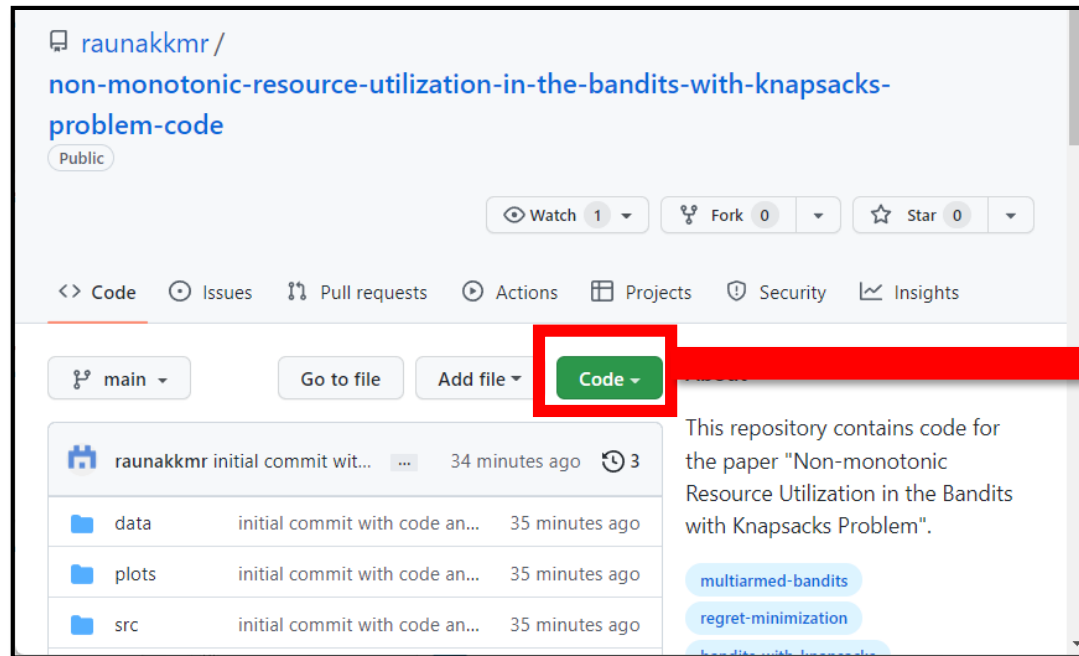
- 원격 저장소를 기존에 연결된 로컬 저장소가 아닌 다른 로컬 저장소에서 사용하기 위해 원격 저장소에 담긴 내용 전체를 로컬 저장소로 가져오는 기능
- 언제 사용하는가?



- ① 나의 원격 저장소를 여러 대의 컴퓨터에 연결하여 사용하는 경우  
(예) 내 원격 저장소를 집 컴퓨터와 사무실 컴퓨터, 학교 컴퓨터 등에 연동하여 사용
- ② 다른 사람의 원격 저장소를 나의 로컬 저장소와 연동하여 사용하는 경우  
※ 이 경우에는 별도의 권한을 부여 받지 않으면 push 명령을 수행할 수 없음.

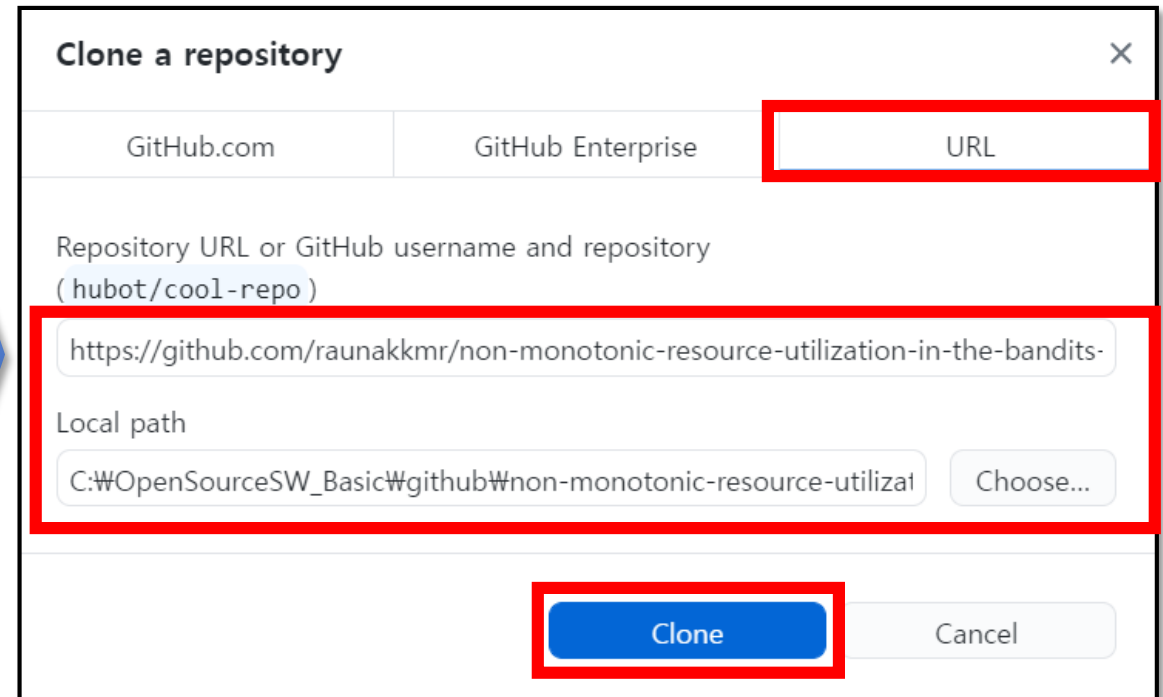
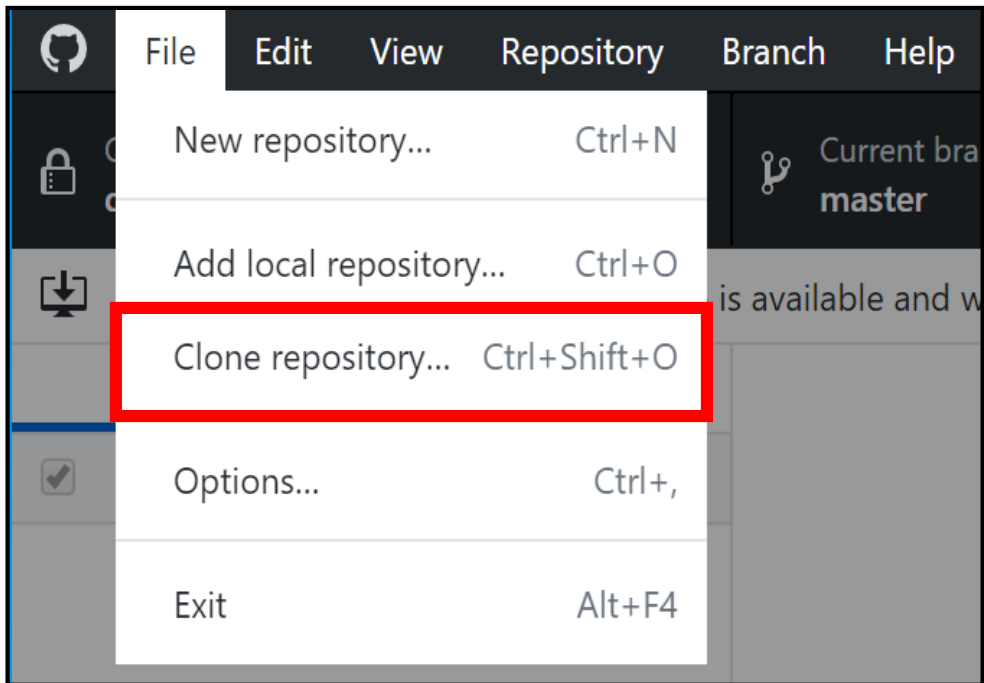
# Clone repository

- GitHub 홈페이지의 “Code” 버튼을 클릭한 후 “Clone”을 선택하면 아래 그림과 같이 링크를 복사할 수 있는 창이 출력되며, 해당 창에서 원격 저장소 주소를 복사(Ctrl+ C)함.  
 ➤ <https://github.com/kyungskim-kit/first-repository> 에 접속하여 직접 실습해 볼 것.



# Clone repository

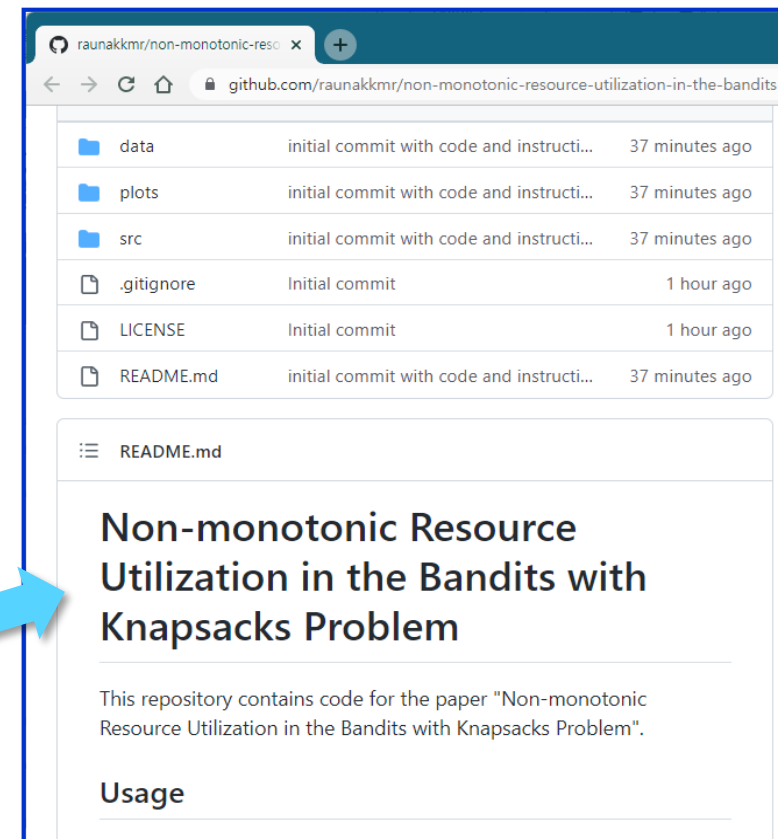
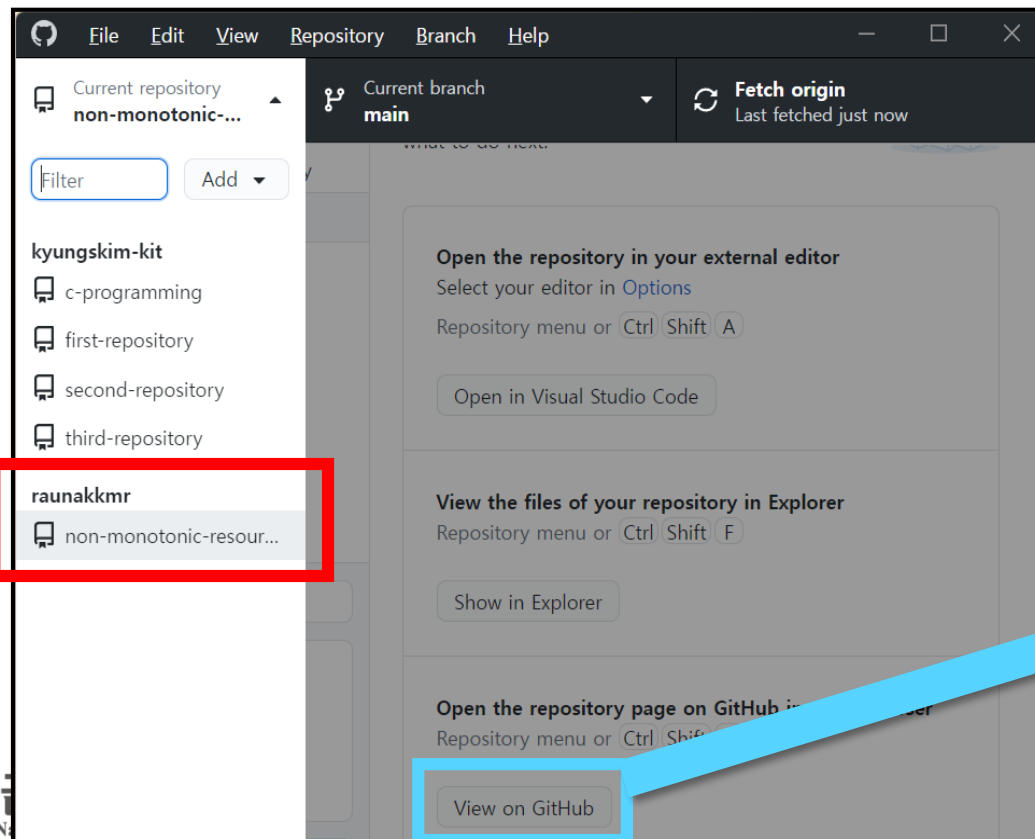
- GitHub Desktop의 “File-Clone repository”를 클릭
- “Clone a repository” 창에서 “URL” 탭을 선택한 후 방금 복사한 원격 저장소의 URL 주소를 붙여넣기(Ctrl+ V)한 후 “Clone” 버튼을 클릭





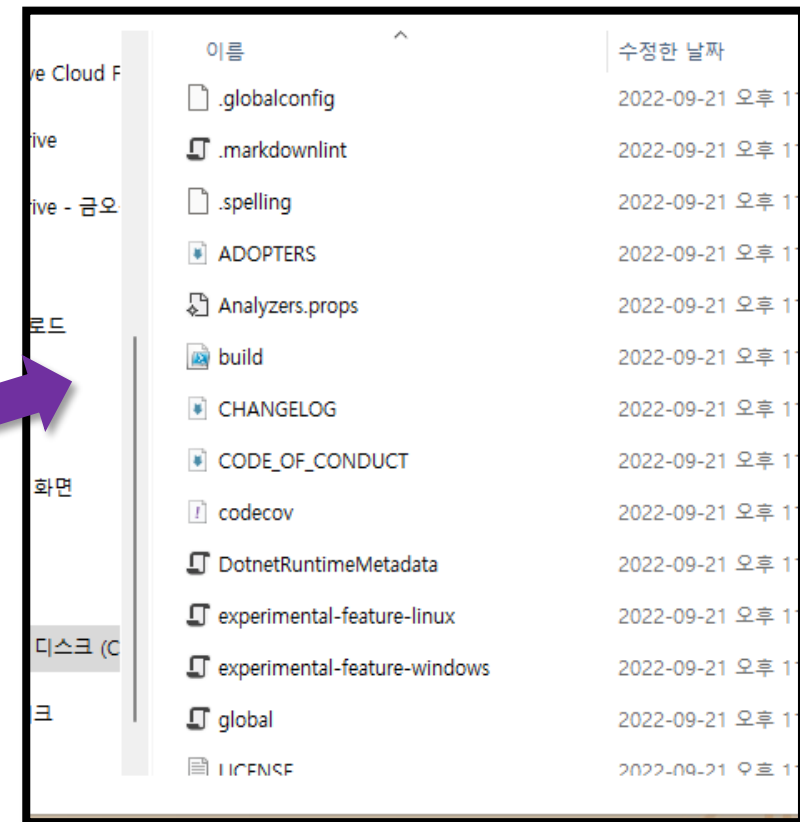
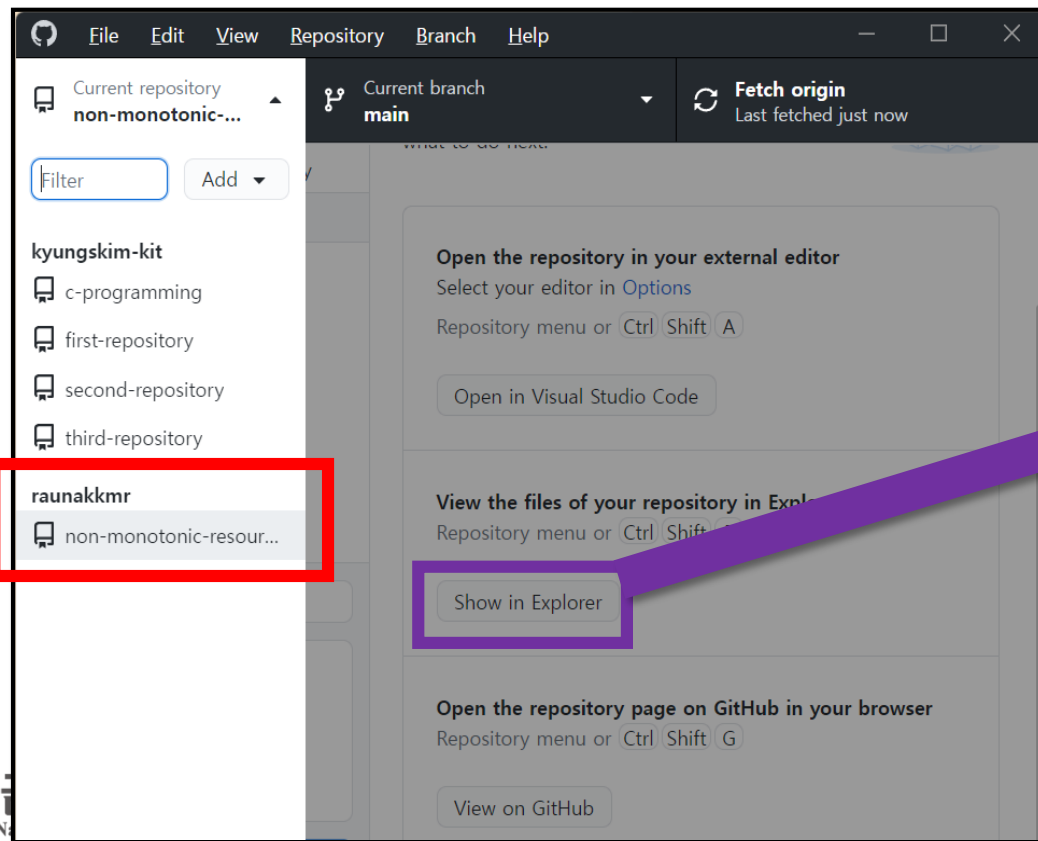
# Clone repository

- 화면 왼쪽 상단의 “Current repository” 또는 “View – Repository list”를 클릭하면 저장소 리스트에 방금 복제한 원격 저장소가 나타난 것을 확인할 수 있음.
- 저장소 리스트에서 방금 복제한 저장소를 선택한 후 “View on GitHub”를 클릭하면 해당 원격 저장소를 확인할 수 있음.



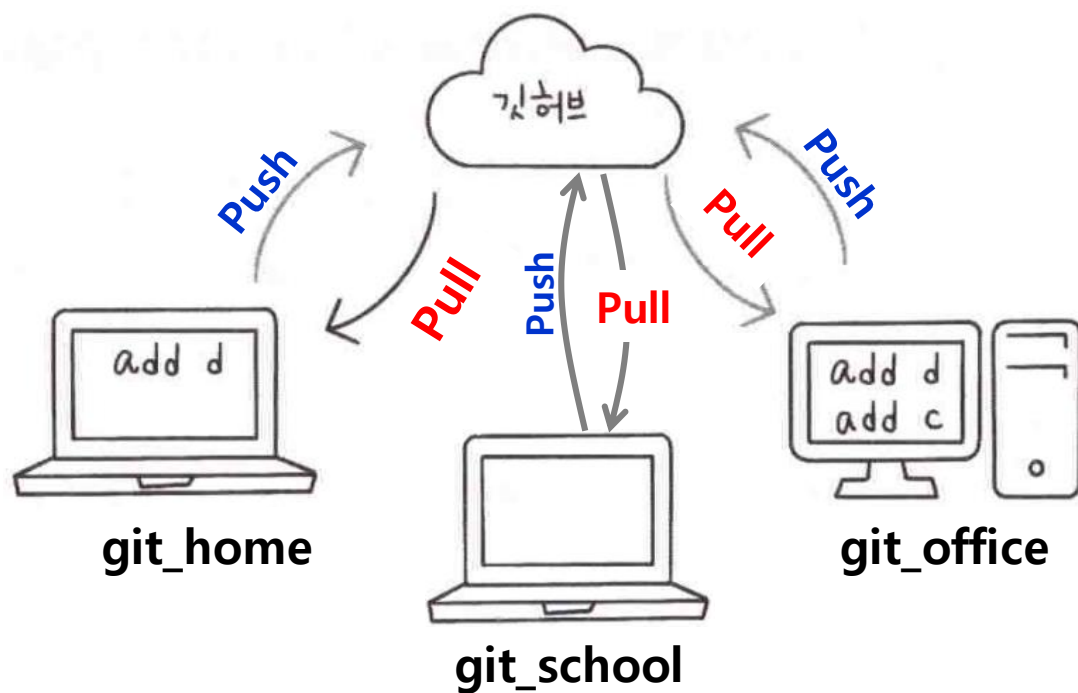
# Clone repository

- 한편, “Show in Explorer”를 클릭하면 해당 복사된 저장소와 파일들을 확인할 수 있음.
  - [주의] 해당 원격 저장소가 내 계정이 아닌 경우 workspace에 복사된 파일들을 수정하여 commit해도 원격 저장소로 push되지 않음. (→ 별도의 권한을 받는 경우에 한해서만 push가 가능)



# Pull repository

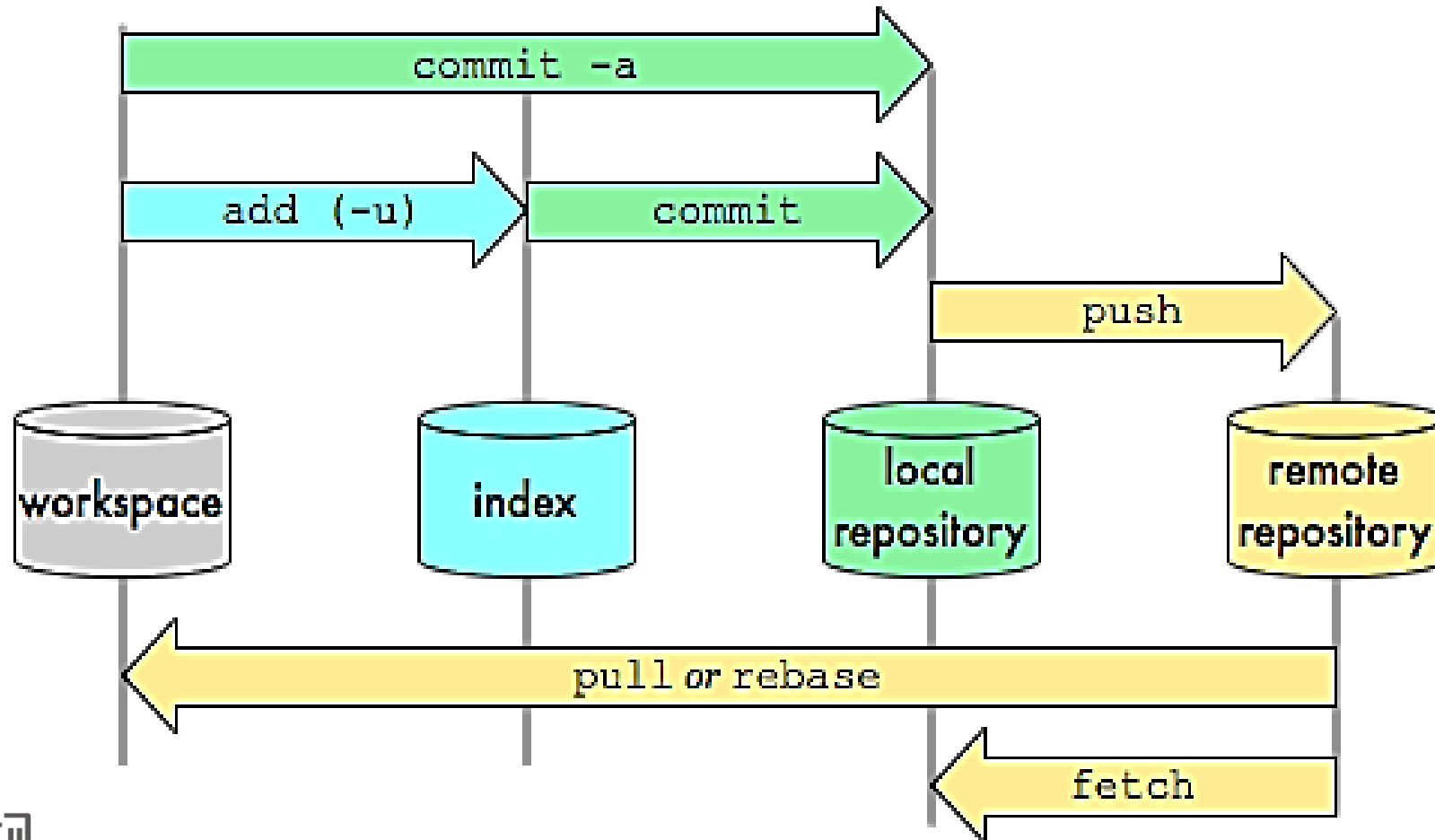
- 원격 저장소의 최신 내용을 연동된 로컬 저장소에 반영하는 기능
  - 원격 저장소 내의 문서들은 다른 로컬에서 수행한 push 명령 등으로 인하여 자주 변동되므로 연동된 원격 저장소의 최신 내용을 주기적으로 로컬 저장소에 반영할 필요가 있음.
- 언제 사용하는가?



- 원격 저장소가 여러 대의 컴퓨터(로컬 저장소)와 연동되어 사용될 때, 특정 로컬 저장소에서 원격 저장소로 push한 경우 다른 로컬 저장소의 내용도 원격 저장소의 최신 상태와 동일하게 내용을 유지하기 위해 사용함.
- 따라서 하나의 원격 저장소에 여러 대의 로컬 저장소를 연동하여 사용하는 경우에는 반드시 push와 pull을 주기적으로 수행해야 함.

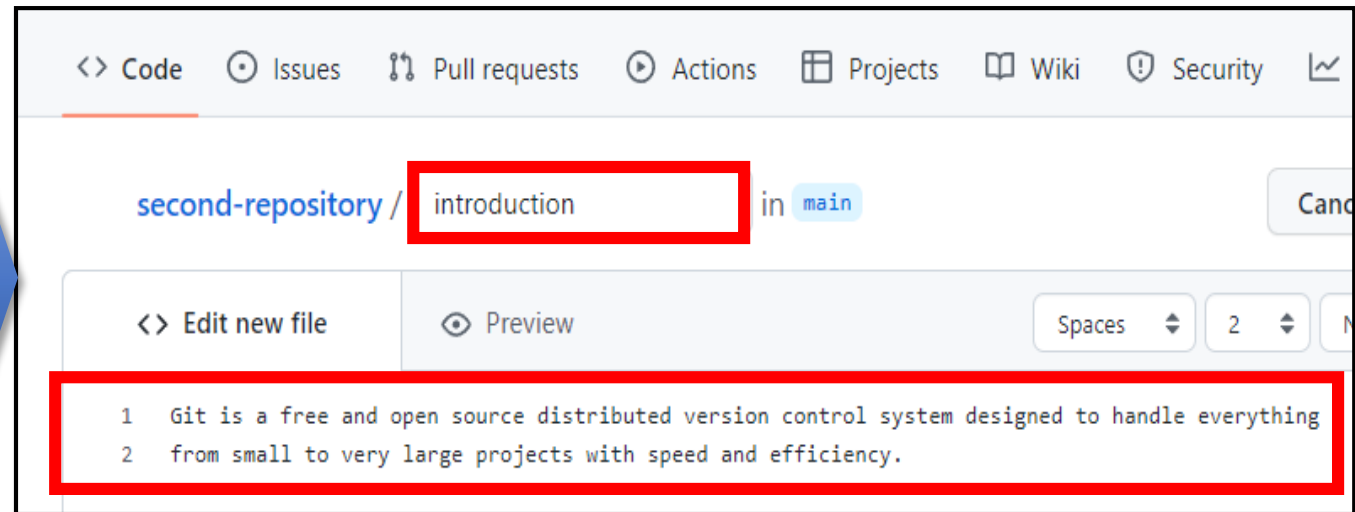
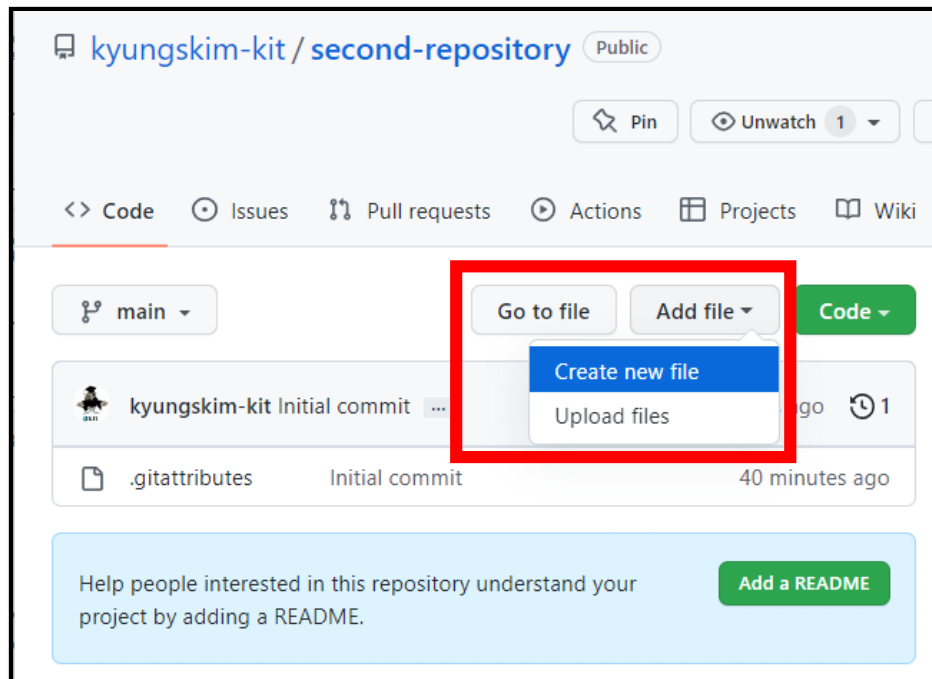
# Pull repository

- Add, Commit, Push, Fetch, and Pull 명령 개념 Review



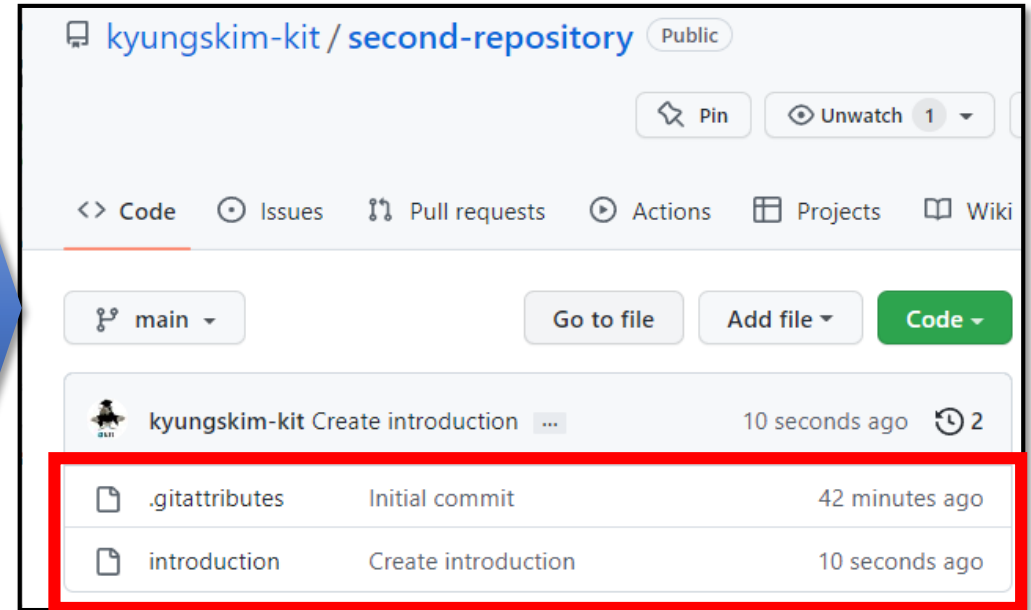
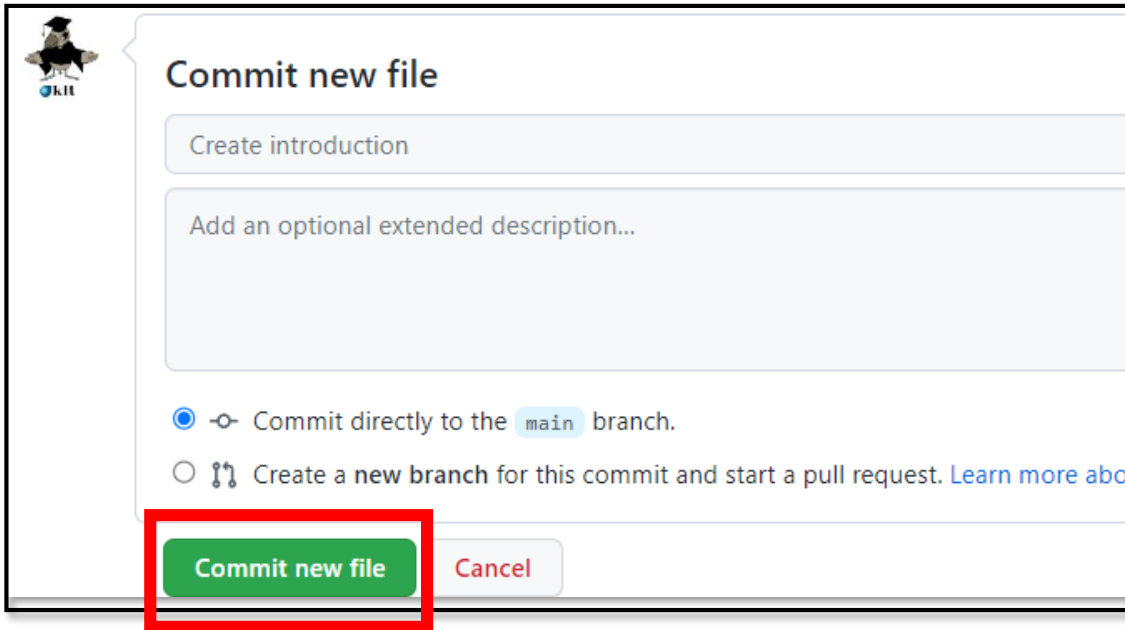
# Pull repository

- 원격 저장소의 문서 내용을 임의로 변경하고 해당 변경 사항을 로컬 저장소에 반영하기
  - 본인의 GitHub 계정에 접속한 후 “second-repository”를 클릭
  - “Add file”을 클릭한 후 “Create new file” 선택
  - 파일명을 “introduction”으로 입력한 후 “<>Edit new file” 탭에 아래와 같이 임의의 내용을 작성



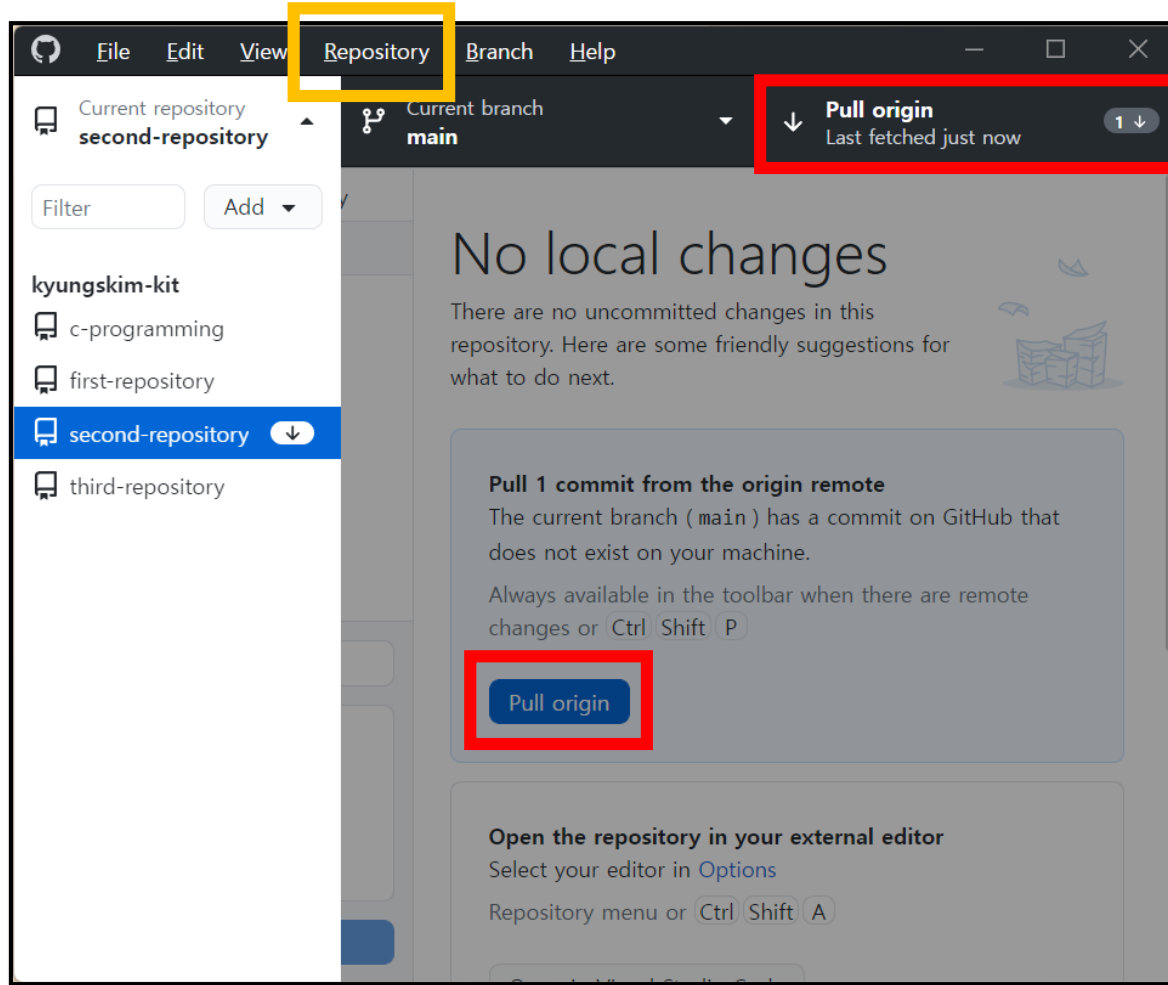
# Pull repository

- 화면 하단의 “Commit new file” 버튼을 클릭하면 방금 작성한 “introduction” 파일이 원격 저장소에 추가되었음을 확인할 수 있음.



File	Commit Message	Time
.gitattributes	Initial commit	42 minutes ago
introduction	Create introduction	10 seconds ago

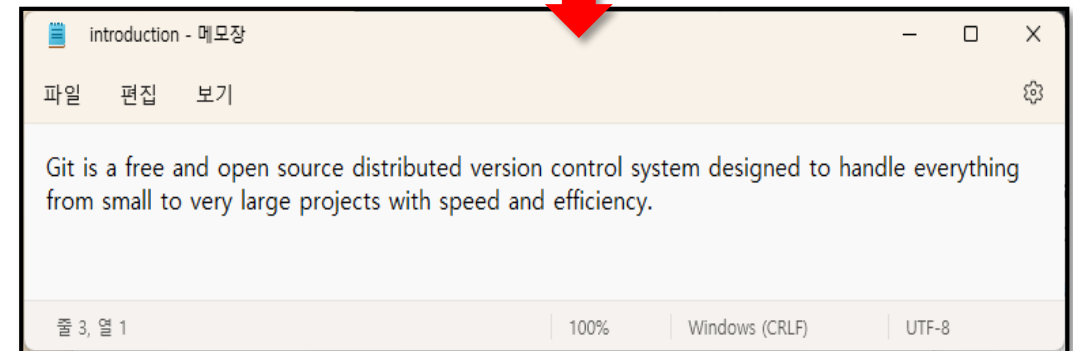
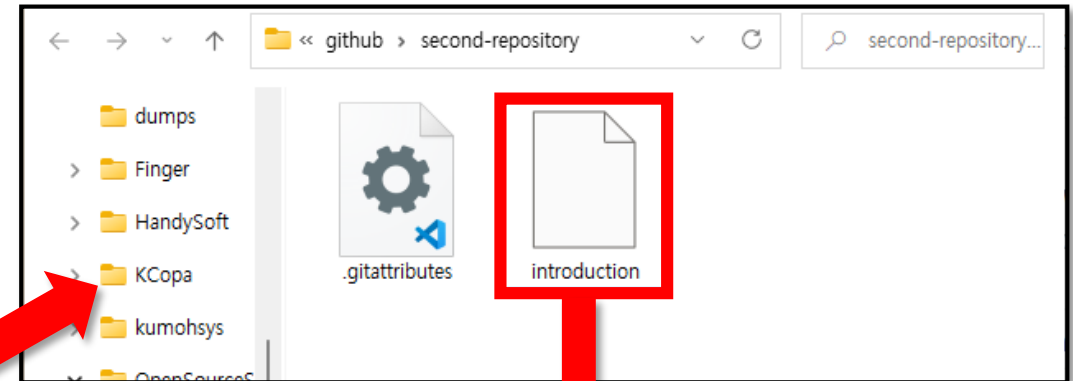
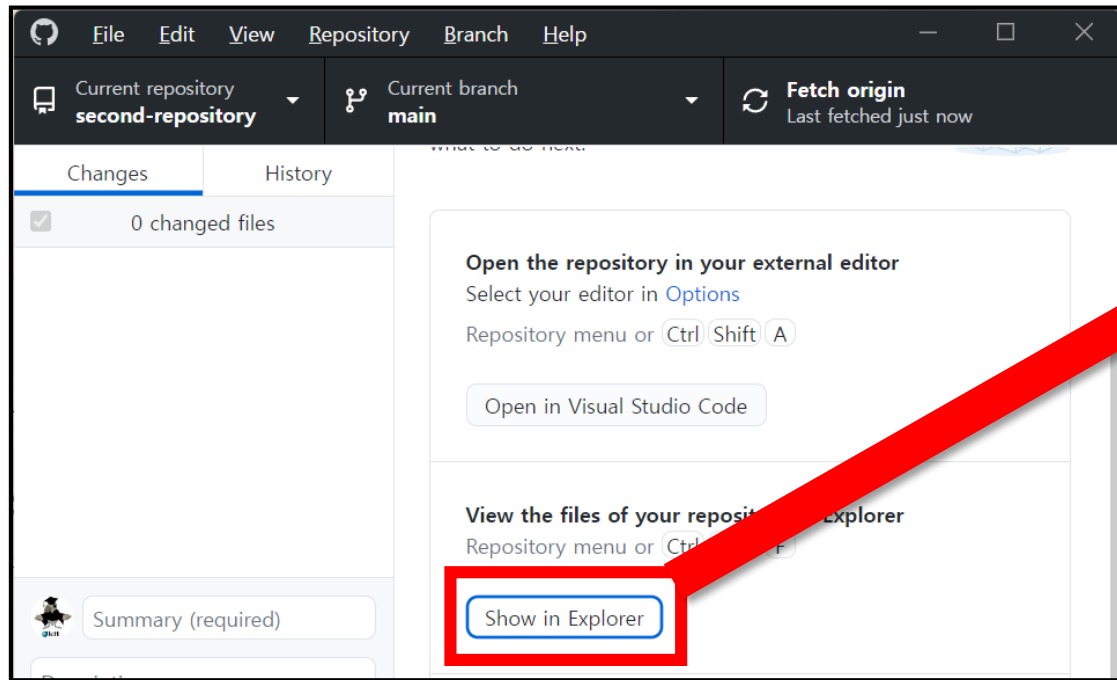
# Pull repository



- GitHub Desktop 화면 왼쪽 상단의 “Current repository”를 클릭한 후 “second-repository” 선택
  - 현재 사용하는 로컬 저장소가 “second-repository”로 변경됨.
- 화면 오른쪽 상단의 “Pull origin” 또는 상단 메뉴바의 “Repository – Pull” 클릭
  - 현재 사용중인 로컬 저장소 “second-repository”와 연동된 원격 저장소의 최신 내용을 가져오는 기능

# Pull repository

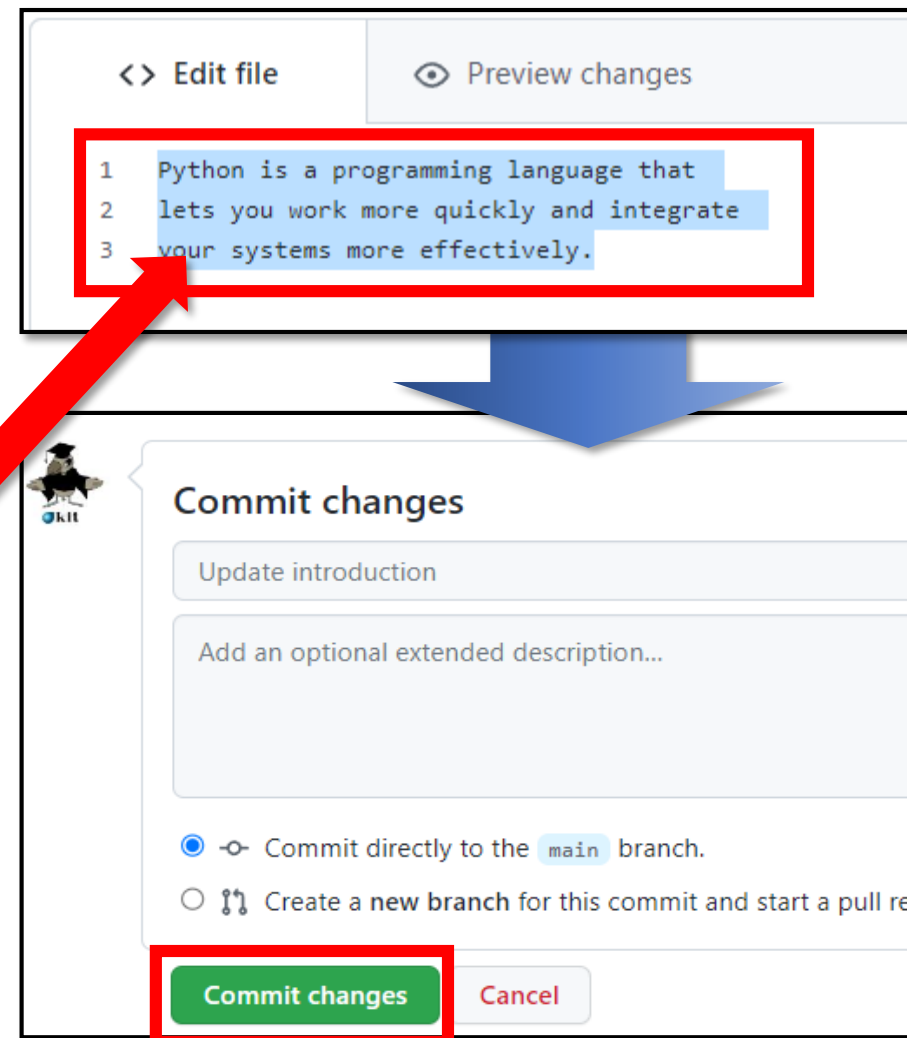
- 오른쪽 화면 하단의 “Show in Explorer” 버튼을 클릭하면 “second-repository” workspace에 “introduction” 파일이 생성되었음을 확인할 수 있음.





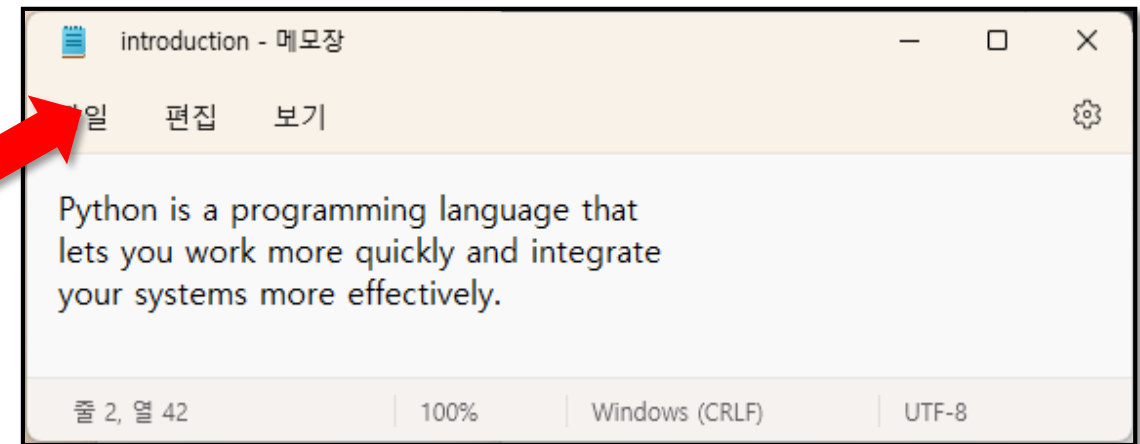
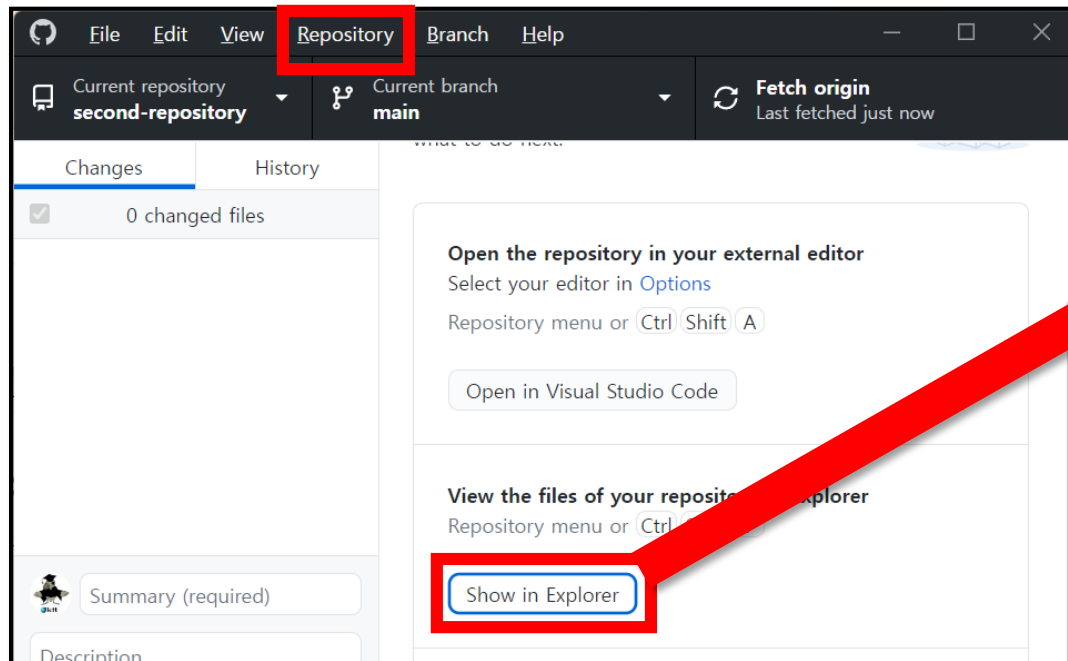
# Pull repository

- GitHub 원격 저장소에 접속하여 방금 생성한 “introduction” 파일의 내용을 오른쪽과 같이 수정한 후 화면 아래의 “Commit changes” 버튼 클릭



# Pull repository

- GitHub Desktop에서 화면 오른쪽 상단에 “Fetch origin”이 “Pull origin”으로 변경되지 않으면, 상단 메뉴 바에서 “Repository – Pull”을 클릭하여 pull 명령을 수행함.
- “Show in Explorer” 버튼 (또는 Repository – Show in Explorer) 을 클릭하여 현재 내 컴퓨터의 workspace를 확인하면 introduction 파일의 내용이 변경된 것을 확인할 수 있음.



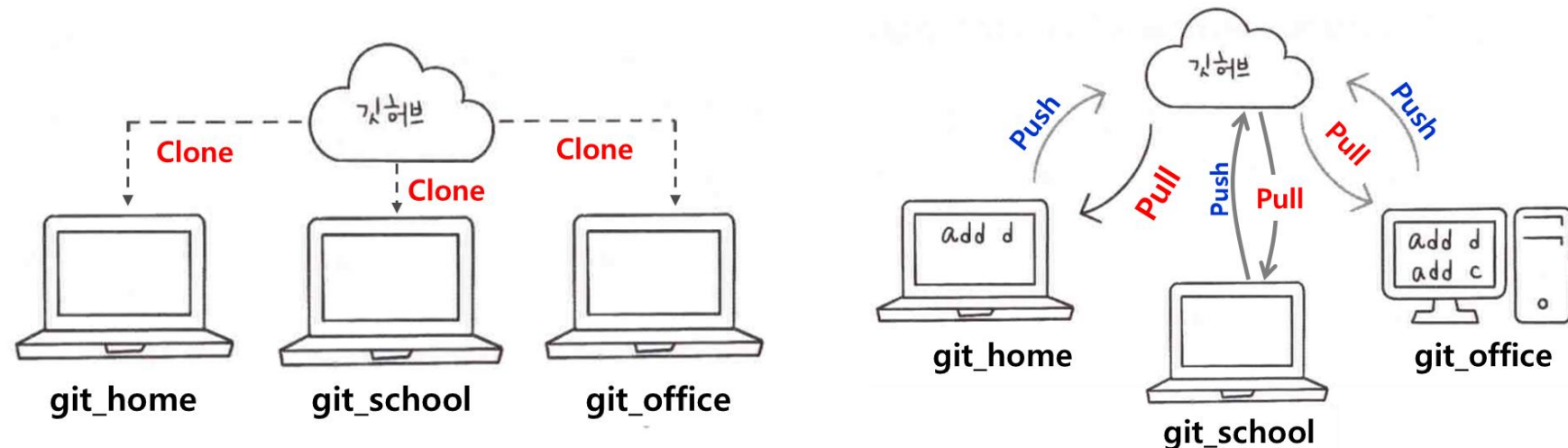
# Clone과 Pull 명령의 기능 비교

## • Clone repository

- 원격 저장소와 연동된 로컬 저장소가 내 컴퓨터에 없는 상태에서 내 컴퓨터에 해당 원격 저장소와 연동된 로컬 저장소를 생성하고 파일들을 복사하여 가져오는 기능
- 즉, 원격 저장소를 복제하여 로컬 컴퓨터에 이식하는 기능

## • Pull repository

- 해당 원격 저장소와 연동된 로컬 저장소가 내 컴퓨터에 있는 상태에서 해당 원격 저장소의 최신 상태를 다운로드하여 로컬 저장소 및 workspace에 반영하는 기능
- 즉, 원격 저장소의 변경 사항을 로컬 컴퓨터에 반영하여 동기화하는 기능



# 종합 실습 I - GitHub Desktop 기초 기능 연습

# 실습 절차 (1/5)

- 1. GitHub Desktop에서 아래의 조건에 따라 로컬 저장소를 생성하시오.
  - 로컬 저장소 생성 경로(폴더): “C:\WOpenSourceSW\_Basic\Wgithub\W”
  - 로컬 저장소 이름: MyNewRepo
  - Readme 파일 자동 생성 여부: No
  - “Git ignore”, “License”가 어떤 종류가 있는지 확인한 후 원하는 옵션을 선택
- 2. 새로 생성된 로컬 저장소의 workspace에 아래의 파일들을 추가하시오.
  - QuickSort.py
  - GCD.py
  - BinarySearch.py

이 파일들은 LMS의 강의자료실에서  
다운받아서 활용하시오.

## 실습 절차 (2/5)

- 3. GitHubDesktop에서 새로 추가된 파일들을 아래의 조건을 참조하여 “Commit”하고 원격 저장소에 “Push”하시오.
  - Commit message는 “My first commit with python codes.”으로 작성할 것.
- 4. 원격 저장소에 접속하여 새롭게 추가한 3개의 파일이 정상적으로 업로드되었는지 확인하시오.

## 실습 절차 (3/5)

- 5. 아래 조건을 참조하여 workspace 내 파일의 내용을 변경하시오.
  - “QuickSort.py” 파일의 7번째 줄에 “# Partition algorithm for the quick sort”이라는 주석과 25번째 줄에 “# Test codes” 주석을 추가하고 저장하시오.
  - “BinarySearch.py” 파일에서 “print” 메소드 내 출력 내용을 다른 내용으로 변경하고, 코드에서 6, 10, 18번째 줄의 주석을 삭제하고 저장하시오.
- 6. 상기 파일들의 변경 사항에 대하여 GitHub Desktop에서 아래의 조건을 참조하여 “Commit”과 “Push” 명령을 순차적으로 수행하시오.
  - Commit message는 “Two files have been modified.”로 작성하시오.
  - 원격 저장소를 Publish할 때, 원격 저장소 이름은 로컬 저장소와 동일하게 “MyNewRepo”로 지정하고, “Keep this code private” 옵션은 해제할 것.

## 실습 절차 (4/5)

- 7. 원격 저장소에 해당 파일들의 수정 사항이 정확하게 반영되었는지 확인하고 수정하기 전과 후의 코드를 비교해본다.
- 8. 원격 저장소에 업로드 된 파일을 원격 저장소에서 아래와 같이 수정한다.  
(※ Workspace 폴더가 아닌 GitHub 사이트에서 수행하는 작업)
  - “BinarySearch.py” 파일을 원격 저장소 상에서 삭제하시오.
  - 원격 저장소 상에서 “GCD.py” 파일의 첫 줄에 주석으로 “# Greatest Common Divisor algorithm”을 작성하고 저장하시오.



## 실습 절차 (5/5)

- 9. GitHub Desktop에서 Fetch & Pull 명령을 수행하시오.
- 10. 원격 저장소 상에서 수정한 내용이 로컬 저장소의 workspace에 정확하게 반영되었는지 확인하시오.

# Appendix – GitHub Desktop 주요 기능 요약

# GitHub Desktop 주요 기능 Summary (1)

기능	관련 메뉴(명령)	부가 설명
시스템에 git 사용자 정보 등록 (※ Git Bash에서 수행)	<code>git config --global user.email "GitHub 계정 이메일"</code> <code>git config --global user.name "GitHub 계정 ID"</code>	
GitHub Desktop 사용자 로그인	File - Options - Accounts	위의 git 사용자 정보 시스템 등록을 먼저 수행한 후 로그인
GitHub Desktop 사용자 계정 확인	File - Options - Git	"Name", "E-mail" 항목에 본인의 GitHub ID와 E-mail 주소 입력
GitHub Desktop 연동 앱 설정	File - Options - Integrations	"Shell", "External editor" 설정 가능 "Shell" 항목은 "Git Bash"로 설정 권장
버전 관리에서 제외할 파일 지정	Repository - Repository settings - Ignored files	내부 설정파일 ".gitignore"를 편집하는 기능 제공

# GitHub Desktop 주요 기능 Summary (2)

기능	관련 메뉴(명령)	부가 설명
새 workspace & 로컬 저장소 생성	File – New Repository	
기존의 로컬 저장소를 импорт	File – Add local repository	기존에 생성되었던 로컬 저장소가 아니면 동작하지 않음.
변경된 문서를 로컬 저장소에 업로드	Commit	여러 문서를 한꺼번에 commit하기 위해서는 “summary”를 반드시 입력하여야 함.
Summary 내용 확인 / 작성	View – Go to summary	
로컬 저장소에 업로드된 변경 문서를 원격 저장소에 업로드	Push origin (또는 Repository – Push)	

# GitHub Desktop 주요 기능 Summary (3)

기능	관련 메뉴(명령)	부가 설명
원격 저장소 전체 내용을 로컬 저장소로 복제	File - Clone	해당 원격 저장소와 연동된 로컬 저장소가 생성/복제됨.
원격 저장소의 최신 내용을 연동된 로컬 저장소에 반영	Repository - Pull	해당 원격 저장소와 연동된 로컬 저장소가 내 컴퓨터에 존재
저장소 리스트 확인	View - Repository list	
저장소 삭제	Repository - Remove	저장소 리스트에서 삭제하려는 저장소를 먼저 선택한 후 remove 명령 수행
Workspace 내 변경된 문서 리스트 확인	View - Changes	
최근까지 수행한 문서 변경 이력 확인	View - History	

# GitHub Desktop 주요 기능 Summary (4)

기능	관련 메뉴(명령)	부가 설명
현재 로컬 저장소 위치에서 Shell (= Git Bash, ...) 열기	Repository – Open in <i>Git Bash</i>	File – Options – Integrations에서 설정된 “Shell”과 “External editor”에 따라 메뉴 이름이 다르게 표시됨.
현재 로컬 저장소 위치에서 external editor (= VS, ...) 열기	Repository – Open in <i>Visual Studio</i>	
현재 로컬 저장소에 대한 workspace 폴더 열기	Repository – Show in Explorer	
현재 로컬 저장소와 연동된 원격 저장소 열기	Repository – View on GitHub	
GitHub Desktop 사용자 가이드 보기	Help – Show User Guides	GitHub Desktop 공식 문서

# Summary

- Git에서는 활용 용도에 따라 하나 이상의 저장소를 운용할 수 있으며, 여러 저장소에 대한 다양한 관리 기능을 제공한다.
- GitHub에서 제공하는 원격 저장소를 다수의 로컬 저장소와 연동함으로써 여러 대의 물리적 장치에서 동일한 환경의 저장소를 구축하고 원격으로 작업을 수행할 수 있다.
- GitHub에서는 저장소의 원격 운용과 복제를 위해 Push와 Pull, Clone 명령을 제공한다.
- 원격 저장소와 연동하여 로컬 저장소를 활용하는 경우 Push와 Pull 명령을 통해 저장소 간 동기화를 수행할 수 있다.

# Q & A