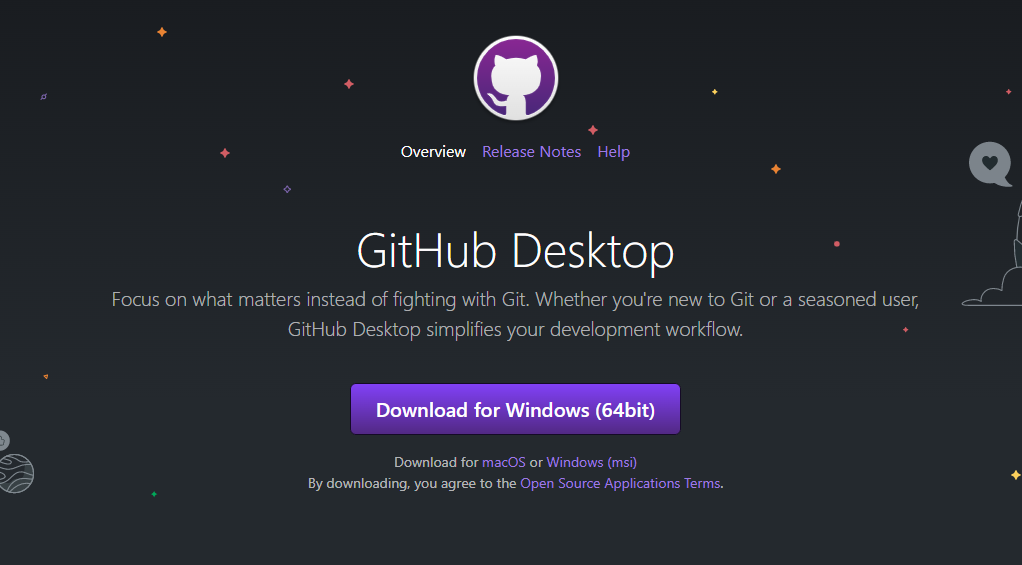
**Github Desktop을 사용하여 Git 저장소와 로컬 저장소를 연결하는 방법**

Hallym univ Kimminseok

1단계: 깃헙 데스크탑 설치 (사진 ctrl + 클릭)

[](https://desktop.github.com/)

2단계: 깃허브 홈페이지에서 새 저장소를 만들거나 기존 저장소를 준비한다.

이때, 새 저장소를 만들 때 Add .gitignore을 선택하고 유니티를 고르면 유니티에서 작업한 내용 중 사소한 변화 등을 무시하고 commit 할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

3단계: 깃허브 홈페이지에서 Code -> Open with Github Desktop으로 깃허브 데스크탑을 실행한다.

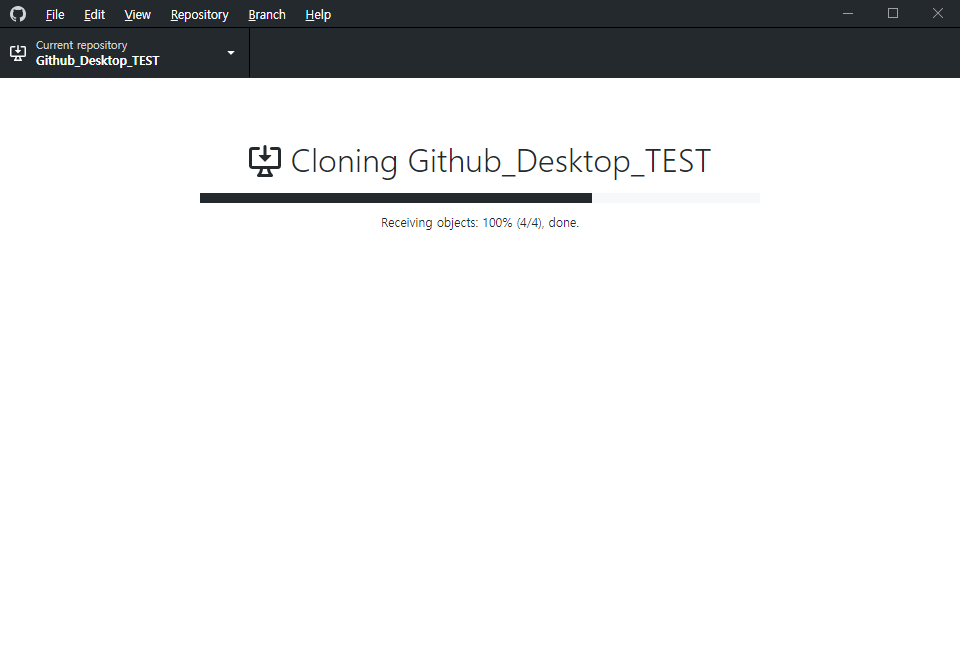
텍스트, 스크린샷, 모니터, 실내이(가) 표시된 사진

자동 생성된 설명

4단계: Git에서 만든 저장소와 연결할 로컬 저장소의 위치를 고르고 Clone을 누른다.

텍스트이(가) 표시된 사진

자동 생성된 설명



이제 로컬 저장소와 Git 저장소 사이에서 파일들을 주고받을 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

**Git 저장소와 로컬 저장소 사이에서 데이터를 주고받는 법**

1단계: Git 저장소에 다른 사람의 파일 추가 등의 변화가 있는지 fetch 누르기

다른 사람이 작업한 것부터 받아야 하는 이유: 여러 사람이 같은 Git 저장소에서 clone을 통해 파일들을 올리다 보면 이름이 같거나 추가, 삭제 등의 작업한 내용이 달라서 파일이 겹칠 때가 있다. 그것을 방지하기 위해 Git 저장소의 변경점을 먼저 받고 나서 수정한 것을 올리는 과정에서 추가적인 수정을 통해 파일이 겹치는 등의 오류가 없게 하기 위해서이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Fetch시 Pull origin이 있다면 Git 저장소에서 추가되거나 삭제 등의 “변경된 점”을 로컬 저장소에 반영한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

Git 저장소에서 직접 추가해준 pull\_test.txt가 로컬 저장소에 저장되었다.

2단계: Github Desktop 왼쪽에 Changes 탭에서 로컬 저장소에 있는 파일들 중에 소스코드의 경우 어느 것이 어떻게 변화되었는지 볼 수 있다.

변경된 로컬 파일들 중에서 변경할 파일만을 선택하여 Commit에 대한 정보(요약, 상세 설명)을 작성하고 Commit to main 버튼을 눌러 Git 저장소에 반영할 준비를 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

3단계: Git 저장소에서 로컬 저장소로 변화된 내용들을 가져올 때 Pull 한 것과 반대로 Push를 통해 로컬 저장소에서 변경된 내용들을 Git 저장소에 적용시킬 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

문제없이 올라갔다면 Git 저장소에서 방금 추가했던 push\_test.txt 파일이 추가된 것을 볼 수 있다.

텍스트, 스크린샷, 모니터이(가) 표시된 사진

자동 생성된 설명