깃허브 Repository에 Pull Request(PR) 넣는 법입니다.

이번 알고리즘 스터디를 하면서 깃허브를 통해 스터디를 진행한 내용을 기록할 예정입니다.

따라서 깃허브와 깃을 사용하는 방법을 알려드리기 위해 해당 문서는 작성되었습니다.

깃(Git)이란?

깃은 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 분산 버전 관리 시스템이다. 라고 위키에 쓰여 있습니다.

쉽게 말해서 협업툴이라고 말할 수 있겠습니다. 어떤 한 프로젝트에 대해 여러 명의 사람이 해당 프로젝트를 관리하고자 할 때 사용하는 툴이라고 생각하시면 편하실 겁니다.

이러한 협업툴을 알고리즘 스터디하는데 왜 사용하나요?

깃에는 commit이라는 기능이 존재하는데요. Commit을 통해 프로젝트 내의 어떤 부분을 변경했는지, 해당 commit이 실행된 시간이 언제인지, 누가 commit을 했는지 알 수 있습니다.

따라서 위의 기능을 이용하여 자신이 어떤 문제를 풀고 발표했는지를 기록할 것입니다.

위의 과정들은 깃허브 내 알고리즘 스터디 Repository내에서 이루어질 것입니다.

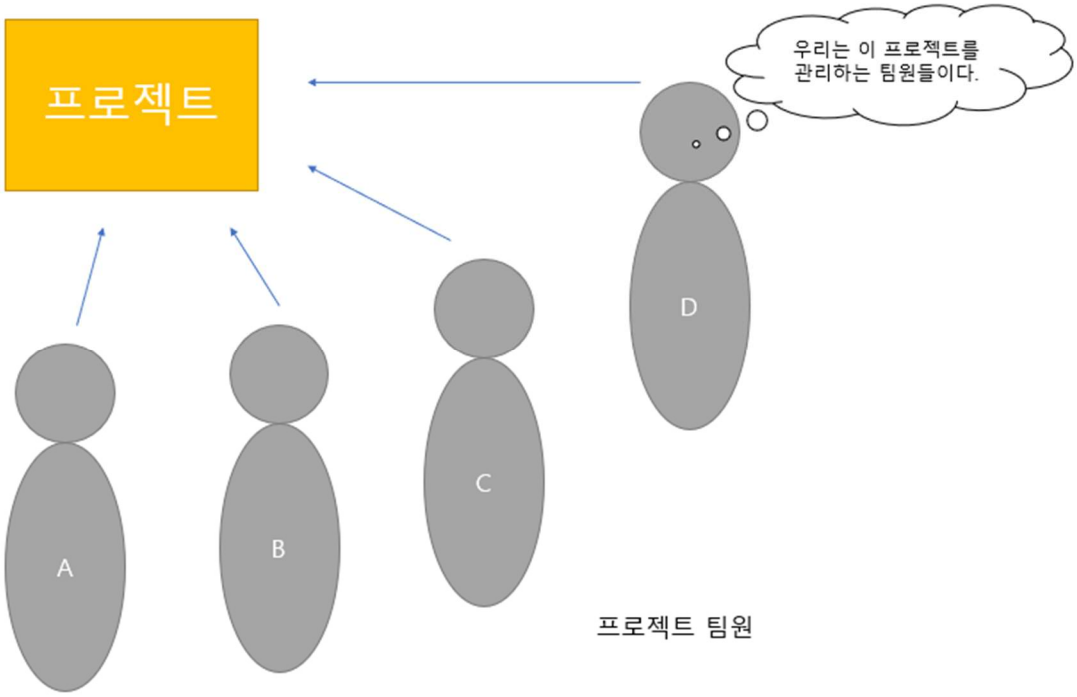

깃허브는 깃의 저장소, 즉 Repository 호스팅을 지원하는 웹 서비스입니다.

깃허브 내 저장소를 이용하면 원격으로 프로젝트의 코드를 관리할 수 있습니다. 본인 PC가 아니더라도 깃허브 저장소만 있으면 어디서든 확인을 할 수 있습니다.

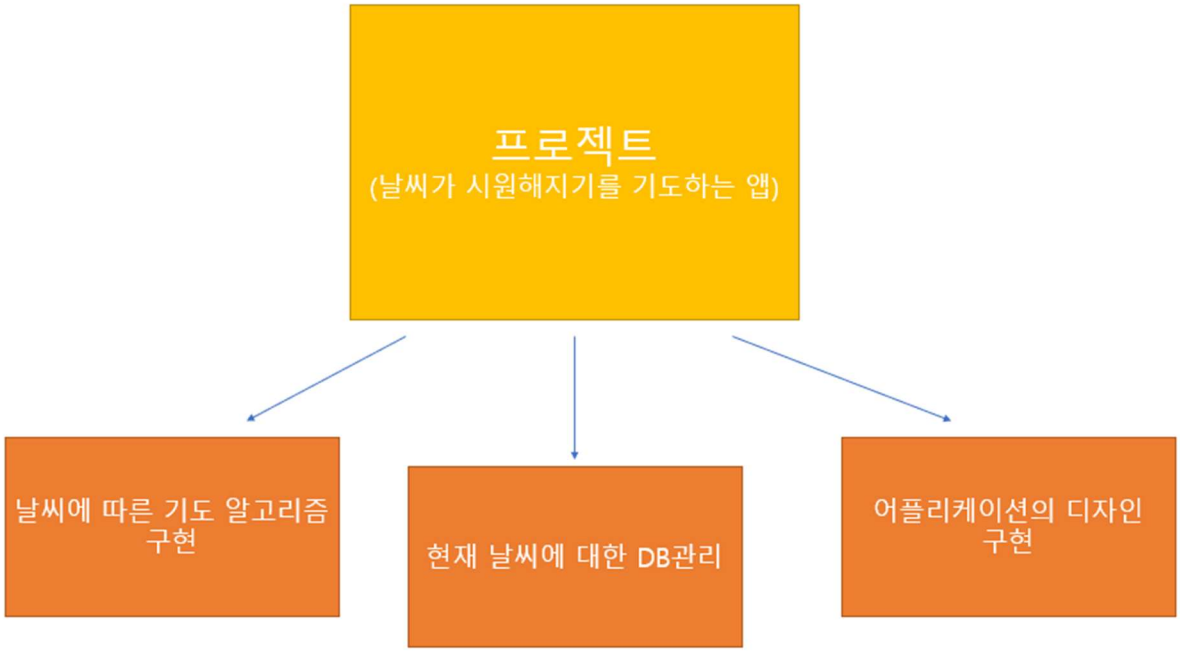
앞서 말씀드렸듯, 우리는 이 Repository에서 알고리즘 스터디 기록을 진행할 것입니다.

사실 위의 설명에서 많은 개념들이 생략되었습니다. 앞으로 Pull Request를 어떻게 하는지 설명하면서 각 개념들에 대한 내용을 설명할 계획이지만, 그렇게 깃에 대한 모든 것을 설명하지는 않을 것입니다. 더 깊은 개념들은 따로 깃을 공부하시는 것을 추천드립니다. 그만큼 복잡한 내용이 많기 때문입니다(간단한 것들은 물어보시면 답해드립니다!).

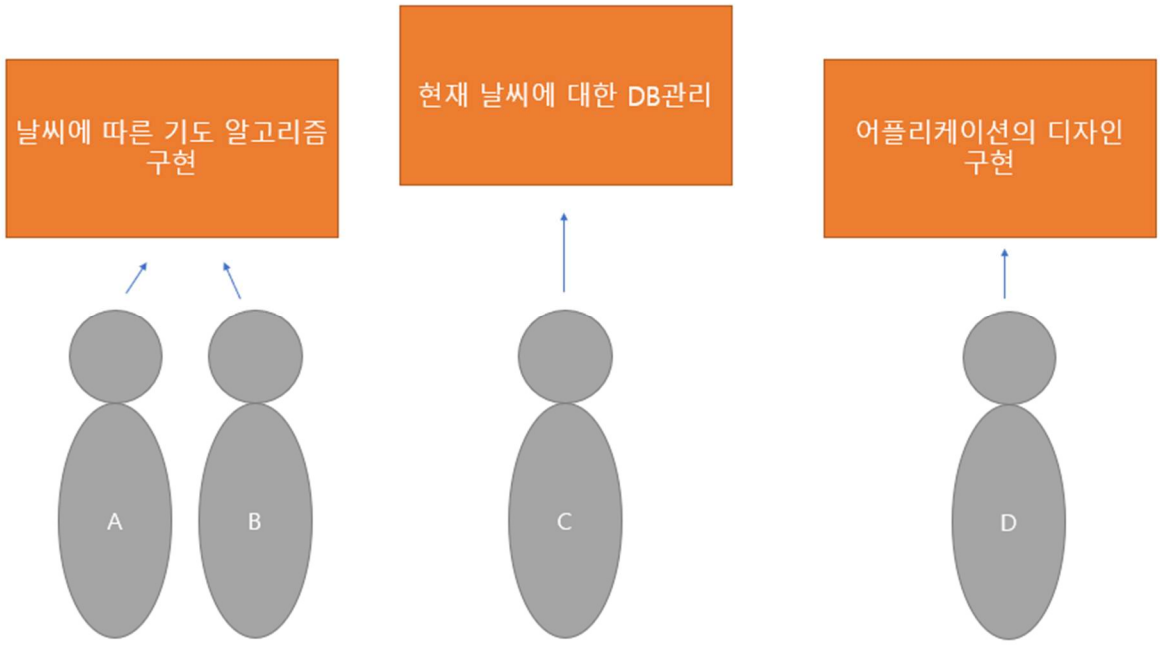
먼저, 깃허브 저장소내에서 프로젝트가 어떻게 관리되는 지부터 설명 드리겠습니다.



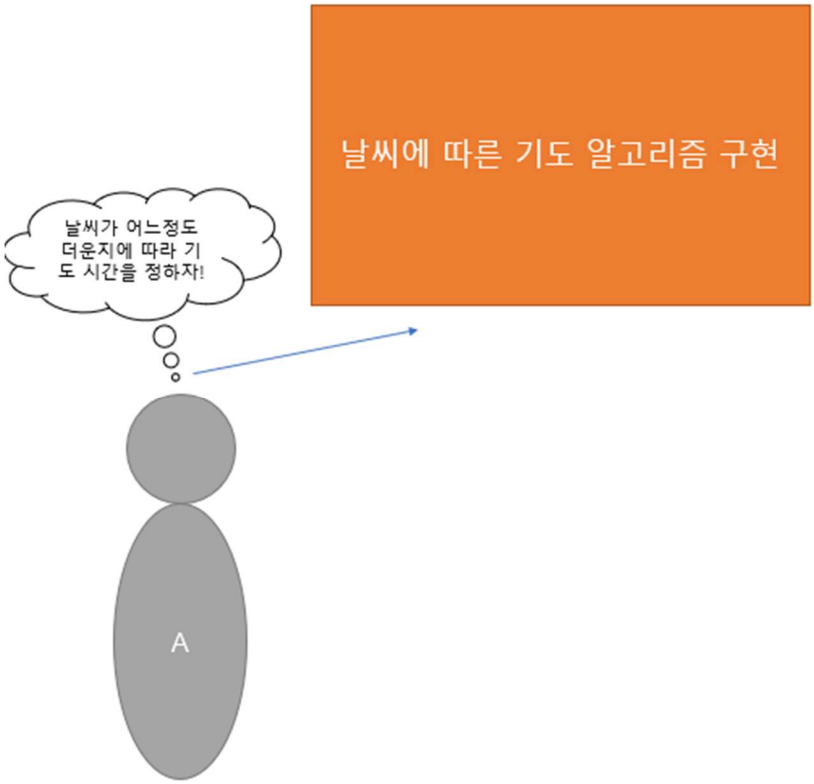

먼저 한 프로젝트를 여러 명의 인원이 협업한다고 가정합시다.



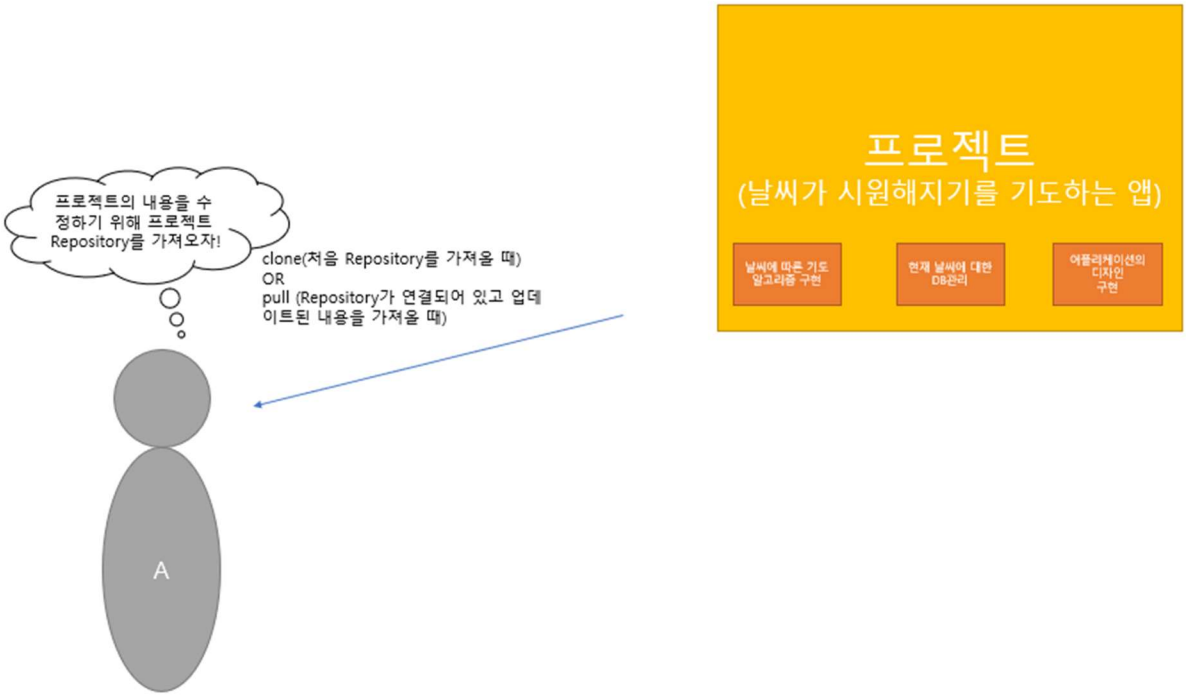

해당 프로젝트의 경우 여러가지 파트로 나눠져서 개발을 할 것입니다.




그러면 해당 파트를 담당하는 인원들이 각 파트에 대한 작업을 할 것입니다.



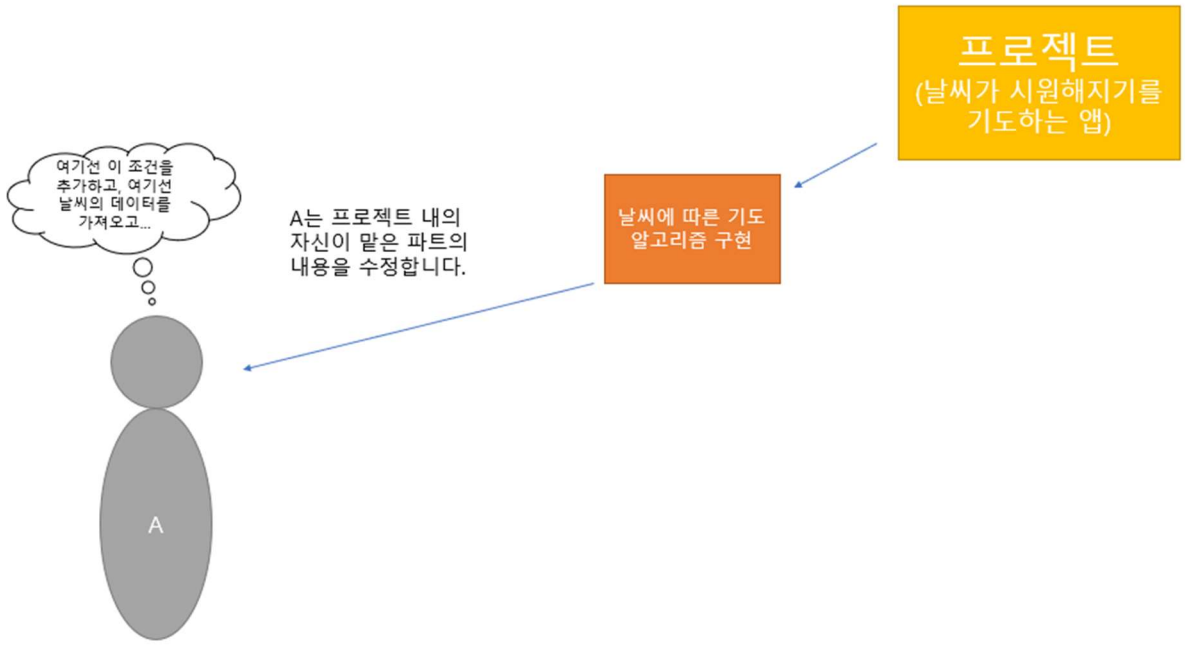

한 담당 인원이 프로젝트를 작업한다고 하면....



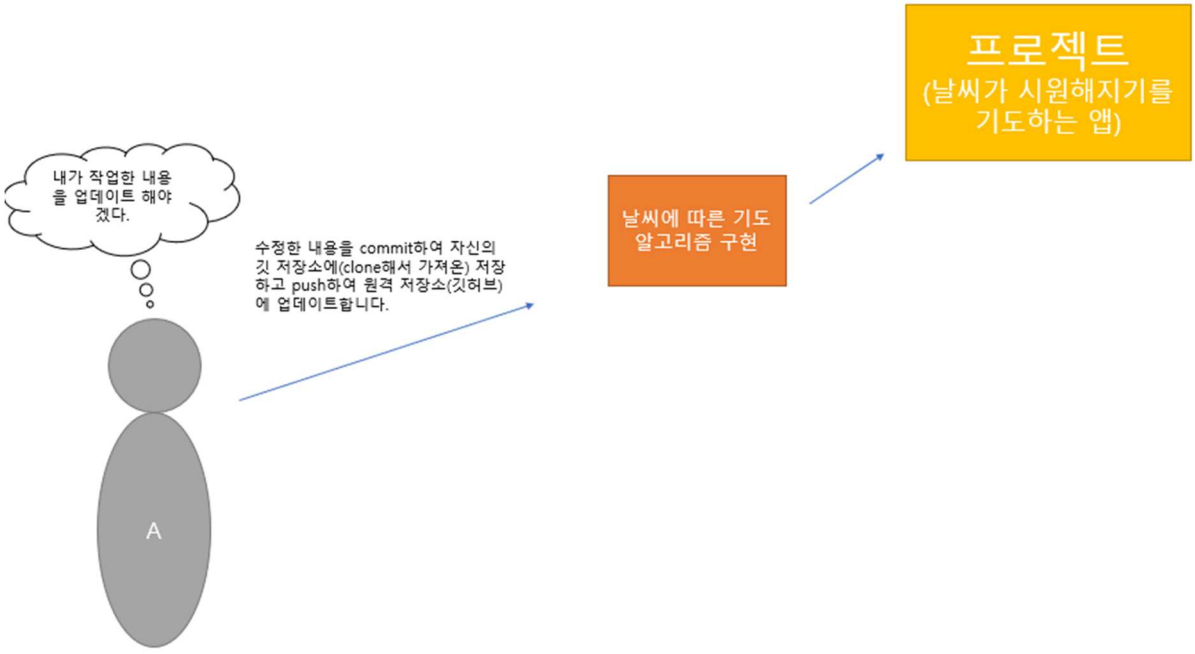

먼저 프로젝트의 데이터를 가져와야겠죠. 깃허브에 있는 Regisotory, 즉 저장소의 데이터를 가져올때 사용하는 명령어가 clone, pull입니다. Clone의 경우 맨 처음 깃 저장소를 가져오는 역할을 합니다. 원래 맨 처음 자기 자신이 깃 저장소를 생성할 때는 init이라는 명령어를 사용하는데요,

협업을 할 경우 깃허브에 존재하는 저장소를 가져오기 때문에 clone을 통해 복제하여 저장소를 가져옵니다.

Pull의 경우 자신이 해당 깃허브의 저장소와 연결이 되어있을 때 사용할 수 있습니다. 연결되어 있을 때 pull을 통해 변경 사항(깃허브의 저장소의 내용)을 자신의 저장소에 업데이트 합니다.

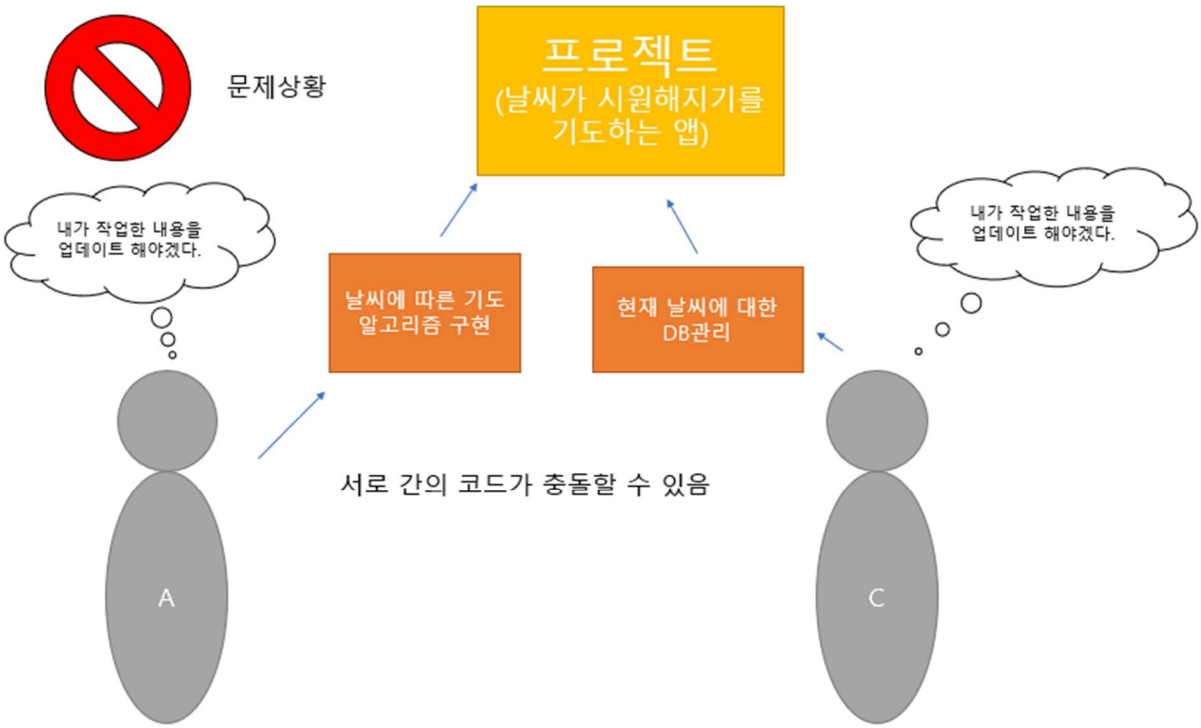



그럼 해당 저장소의 내용을 가지고 작업을 해야겠죠. 자신이 짠 코드를 추가하거나 기존의 코드를 수정할 것입니다(추가적인 내용이지만 코드에만 국한되지 않습니다. 저장소내 모든 내용 포함입니다).

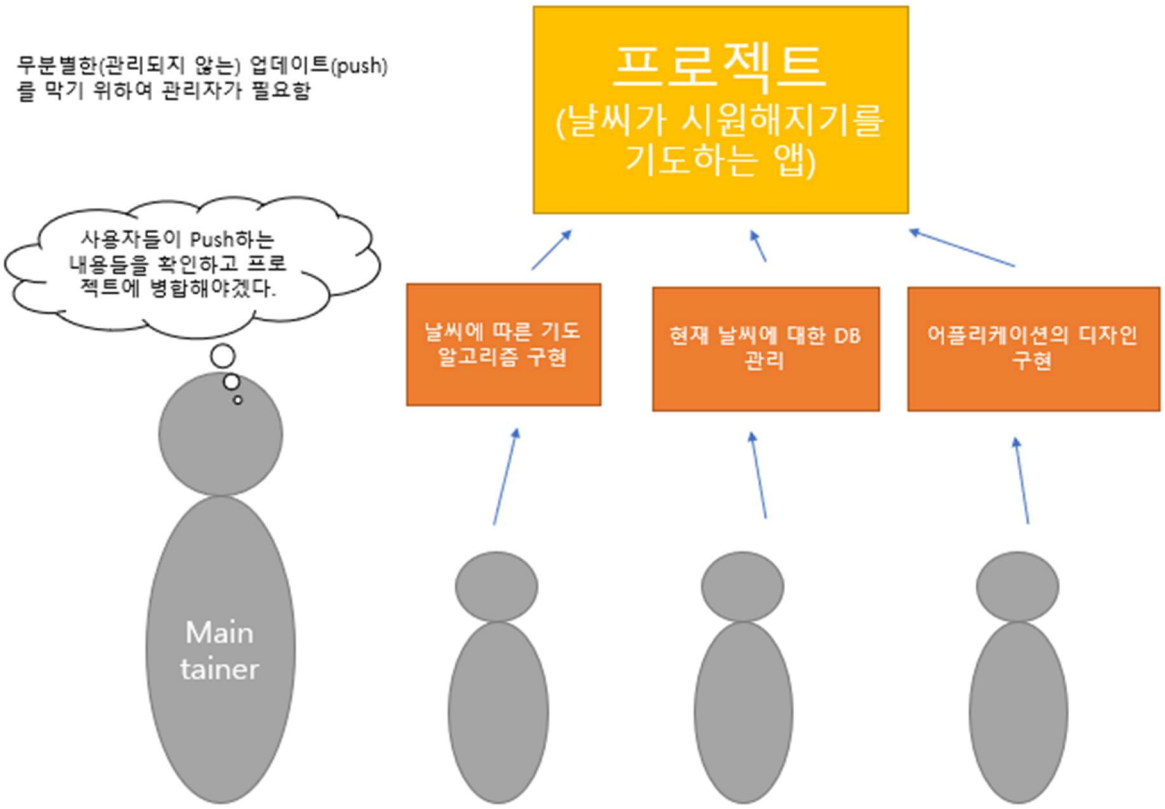


그러면 자신이 작업한 내용을 업데이트 하여 깃허브의 저장소에 저장해야겠죠?

그러기 위해서는 자신의 저장소의 변경사항을 저장해야 합니다. 이를 commit이라고 합니다. Commit을 통해 자신의 깃 저장소의 변경사항을 저장합니다. 그리고 해당 commit내용을 깃허브의 저장소에 업데이트합니다. 이를 push라고 합니다. 여기서 작업 내용 중에 Branch내용을 생략하였습니다. 해당 내용은 밑에 어떻게 Pull Request를 하는지에 대해 설명할 때 하도록 하겠습니다.

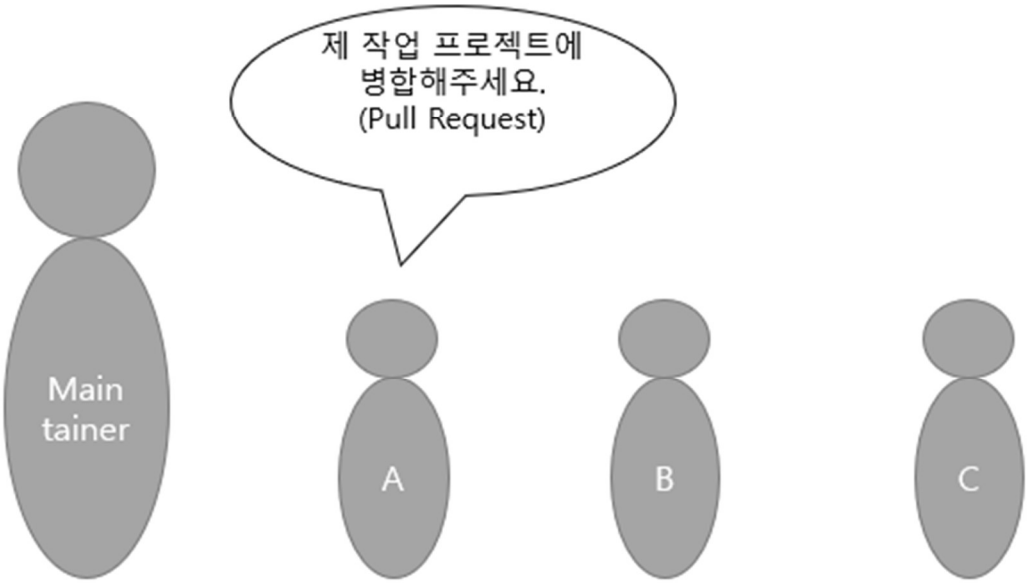



이러한 과정이 혼자 작업한다면 문제가 생기는 경우는 거의 없을 것입니다. 하지만 다른 사람과 같이 작업 중이라면? 서로 작업한 내용물 간의 충돌이 생길 수 있습니다. 변수명을 바꿨는데 해당 변수를 사용중이거나 함수의 내용이 바뀌었다든가 하는 부분이죠.



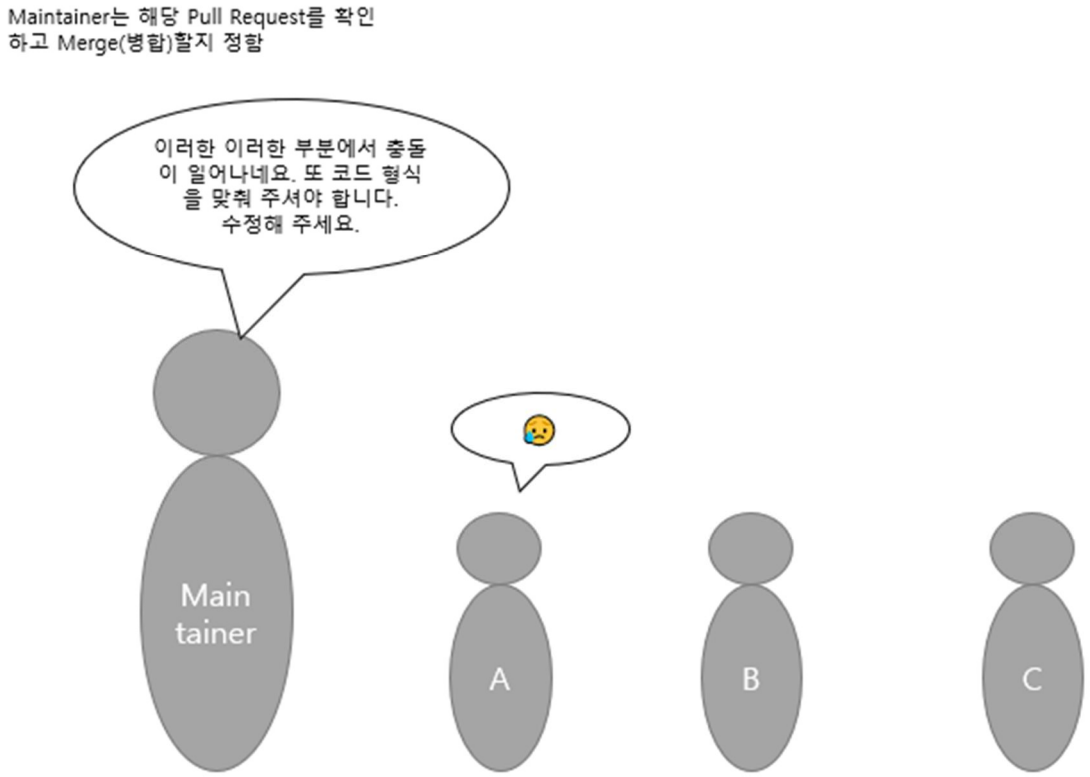

그래서 이를 관리하는 부분이 필요합니다. 프로젝트의 관리자를 Maintainer라고 합니다. 이 Maintainer는 각 사람들이 작업한 내용을 확인하고 해당 내용이 프로젝트에 Merge(병합)해도 괜찮은 지 확인하는 작업을 합니다.

사용자들은 자신의 작업물을 업데이트
하기 위하여 Pull Request를 보냄



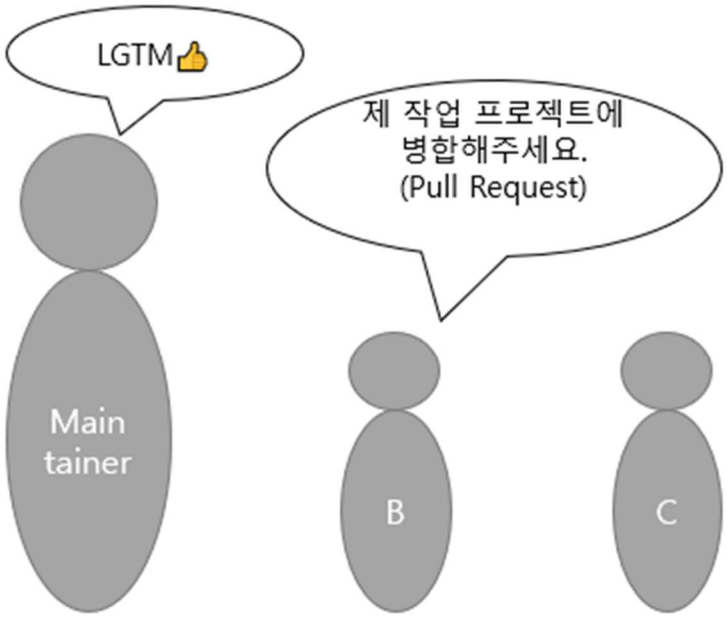

각 사용자들은 자신이 작업한 내용을 프로젝트에 반영하기 위해서 Pull Request를 보냅니다. 여기서 Push Request가 맞지 않냐고 생각하실 수 있습니다. 하지만 여기서 받아들이는 부분이 프로젝트 즉 주체가 프로젝트이므로 변경 사항을 받아들인다고 생각하여 Pull이라고 생각하시면 되겠습니다.

추가적인 설명을 드리자면, 누구나 참여할 수 있는 오픈소스 프로젝트의 경우 참여하는 사람을 Contributor라고 지칭합니다. 각 오픈소스 문서에는 contribute문서가 있고 해당 과정을 안내하기도 합니다.




그러면 Maintainer는 해당 내용을 확인하여 해당 내용이 프로젝트에 Merge가 되어도 되는지 확인합니다.

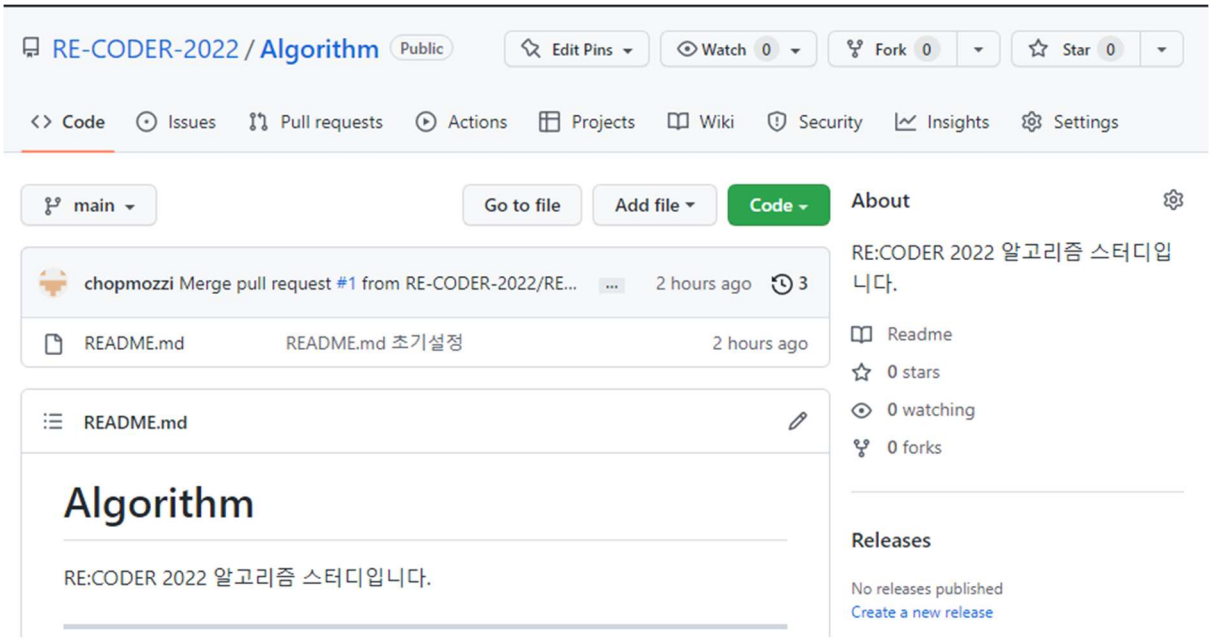
Maintainer는 해당 Pull Request를 확인
하고 Merge(병합)할지 정함




작업한 내용이 프로젝트에 Merge가 되어도 괜찮다고 판단이 들면 해당 내용을 Merge하여 프로젝트에 병합시킵니다.

위와 같은 과정을 통해 깃허브 저장소에서 프로젝트가 관리됩니다.

이게 알고리즘 스터디와 무슨 상관인지 이제 설명하도록 하겠습니다.



우리는 Algorithm 스터디를 위한 Repository, 즉 저장소를 만들어 운영할 것입니다.

위의 Repository내에서 자신이 발표한 알고리즘 문제들을 작성하여 기록하고 업데이트할 것입니다. 그리하여 자신이 어떤 내용을 발표했는지, 얼마나 꾸준히 하였는지 등을 확인할 수 있습니다.

따라서 Repository내에서 자신만의 폴더를 만들고 해당 폴더 내에 자신이 발표한 내용들을 업데이트 하면서 스터디 활동을 기록해 나갈 것입니다.

이제 어떻게 Pull Request를 하는지에 관해 설명하겠습니다.

이번 스터디에서 활용한 방법은 크게 두가지입니다.

1. GithubDesktop을 이용한 깃 사용
2. 깃허브 홈페이지내에서 직접 수정

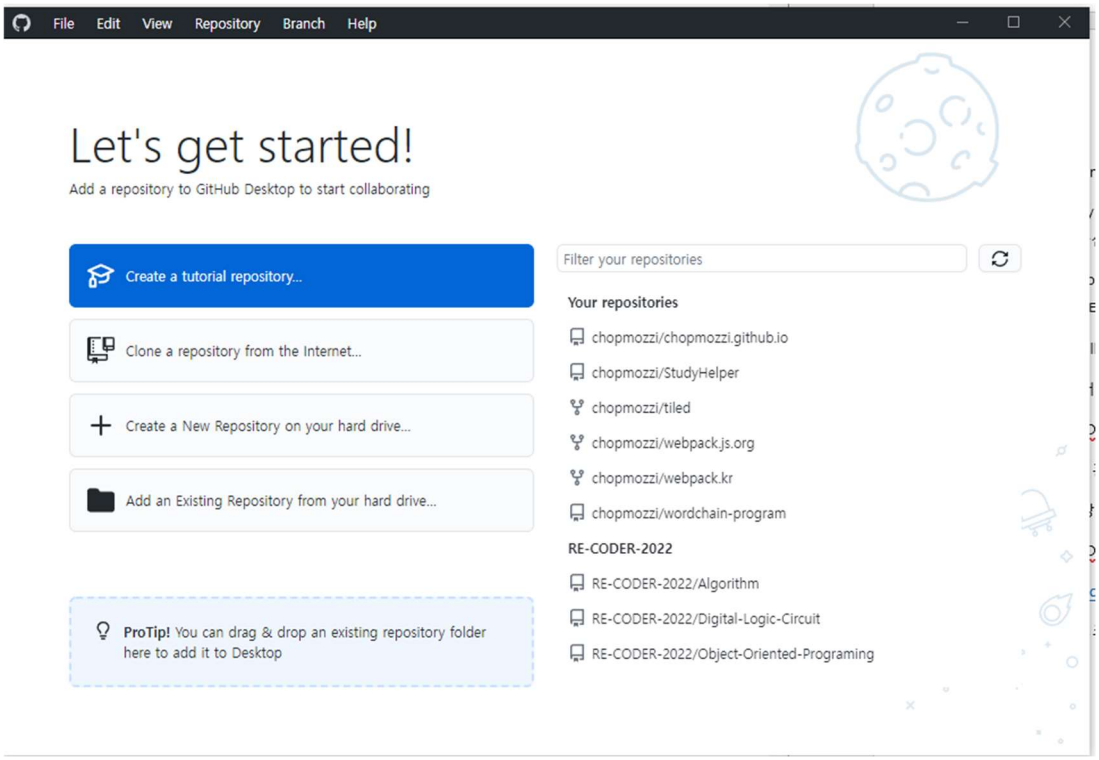
위의 두 가지 방법 중 본인이 원하는 방식을 활용하여 Pull Request를 해주시면 되겠습니다.

1. GithubDesktop을 이용한 깃 사용

   https://desktop.github.com/

   해당 링크에 들어가셔서 설치해주시면 됩니다.

   설치 시 자신의 깃허브 아이디로 로그인해주시고 연동 등의 설정을 완료해주시면 됩니다.



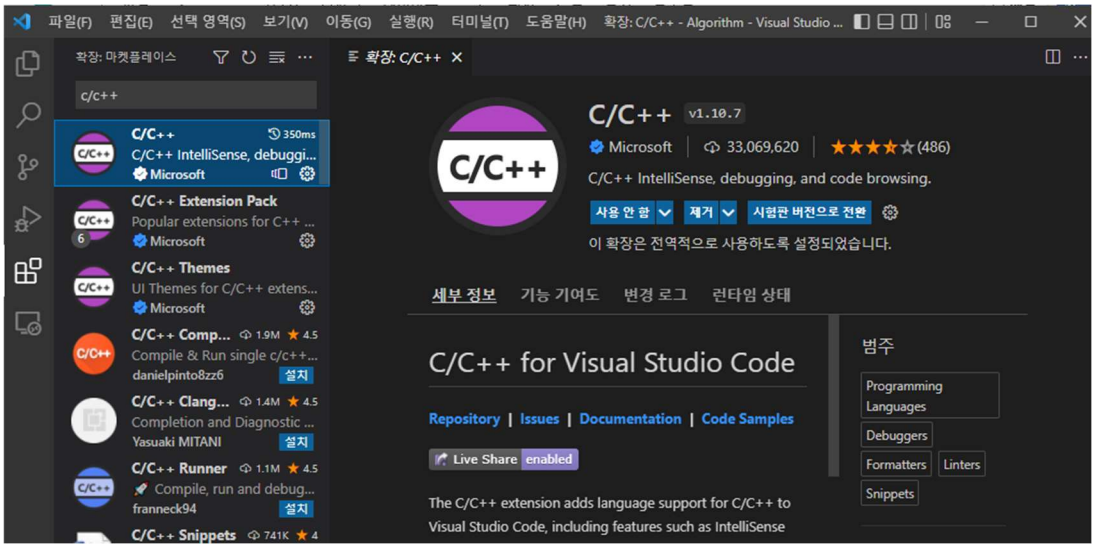

   그러면 위와 같은 화면이 나옵니다.

그 다음은 VSCode를 설치할 것입니다.

https://code.visualstudio.com/download

위의 링크에서 받아 주시면 됩니다.

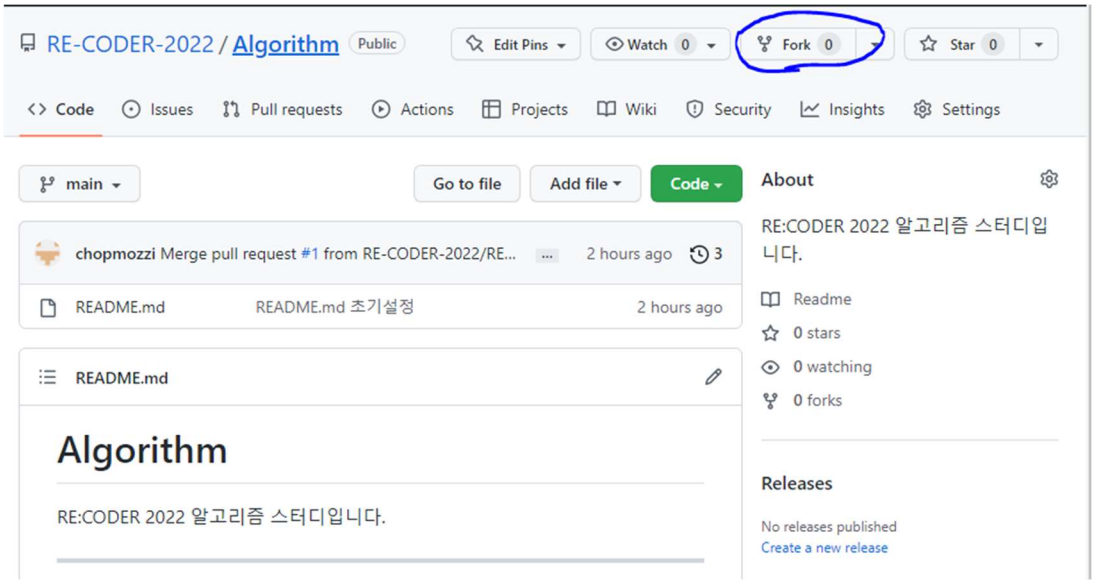
설치해주신다음 Extension 탭에서 Korean, 자신이 사용할 언어팩을 설치해주시면 되겠습니다.



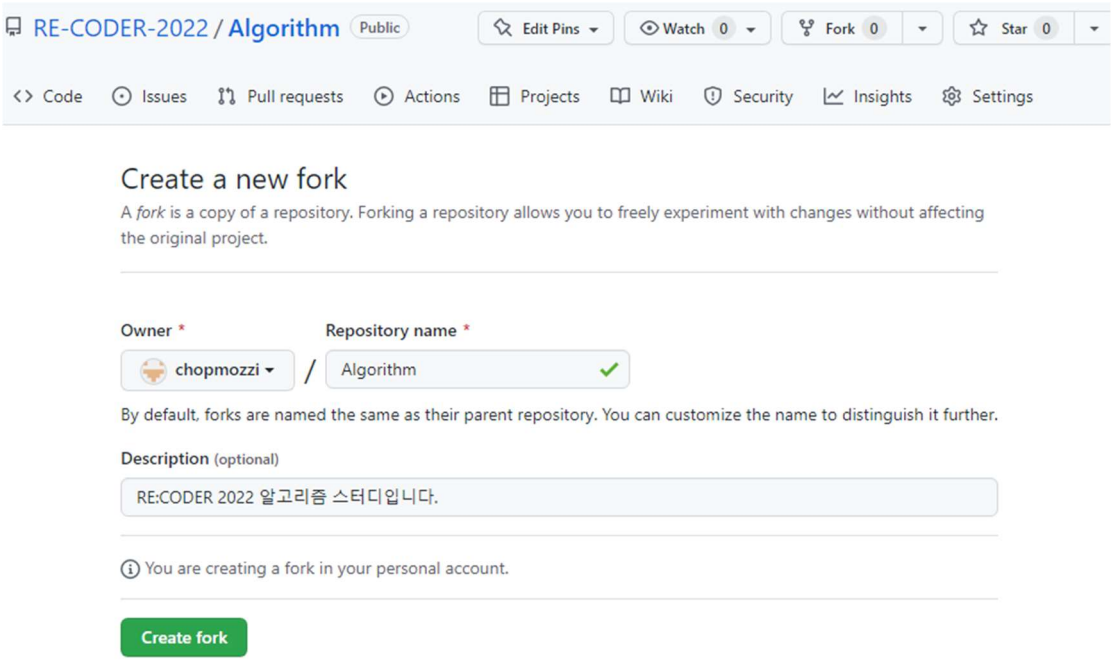

이런 식으로 검색해서 설치해주시면 됩니다.

https://github.com/RE-CODER-2022/Algorithm

그 다음 위의 Repository 주소에서 자신의 Repository로 fork를 할 것입니다.

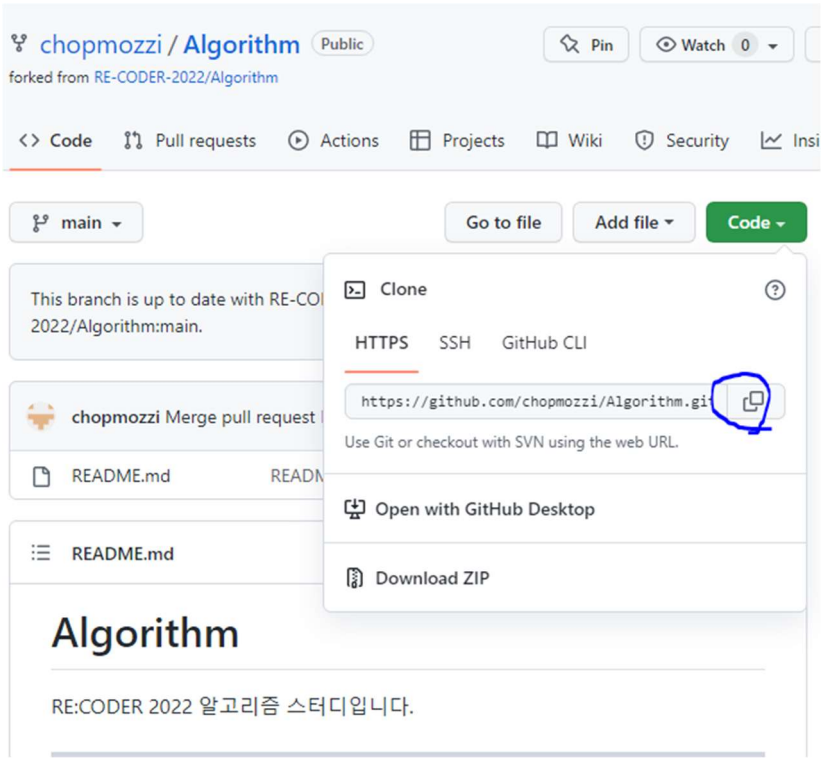



해당 fork 버튼을 눌러줍시다.

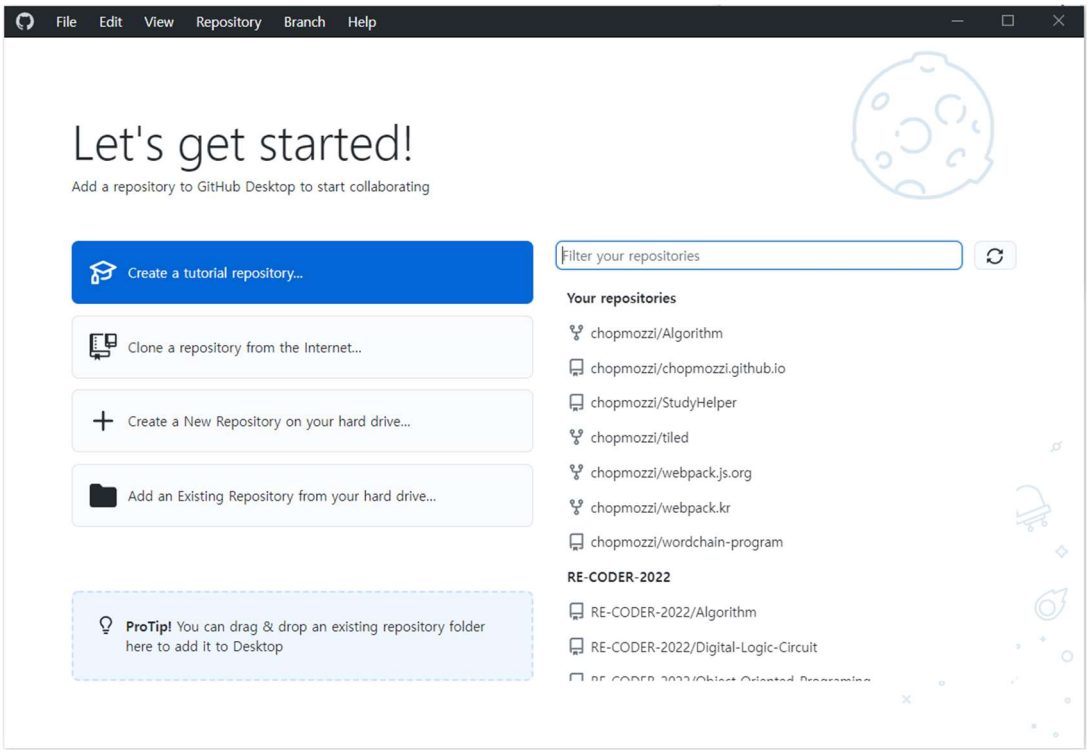



그럼 위와 같은 화면이 나옵니다. 그대로 Create fork를 눌러줍시다.

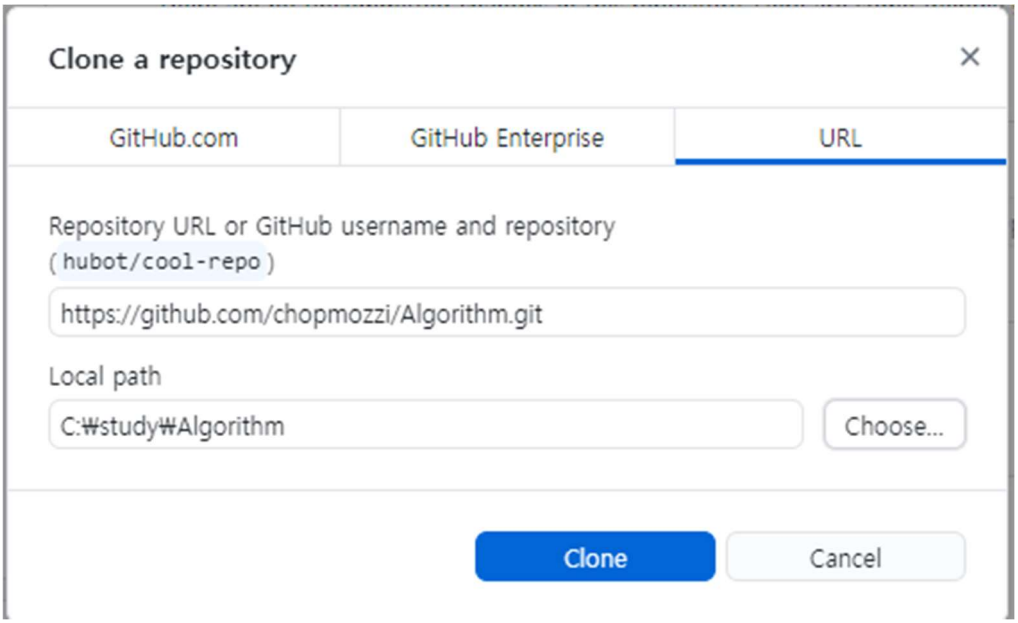
그러면 자신의 계정의 Repository로 그대로 fork가 될 것입니다.




그러면 fork한 Repository에서 URL을 복사해줍시다.

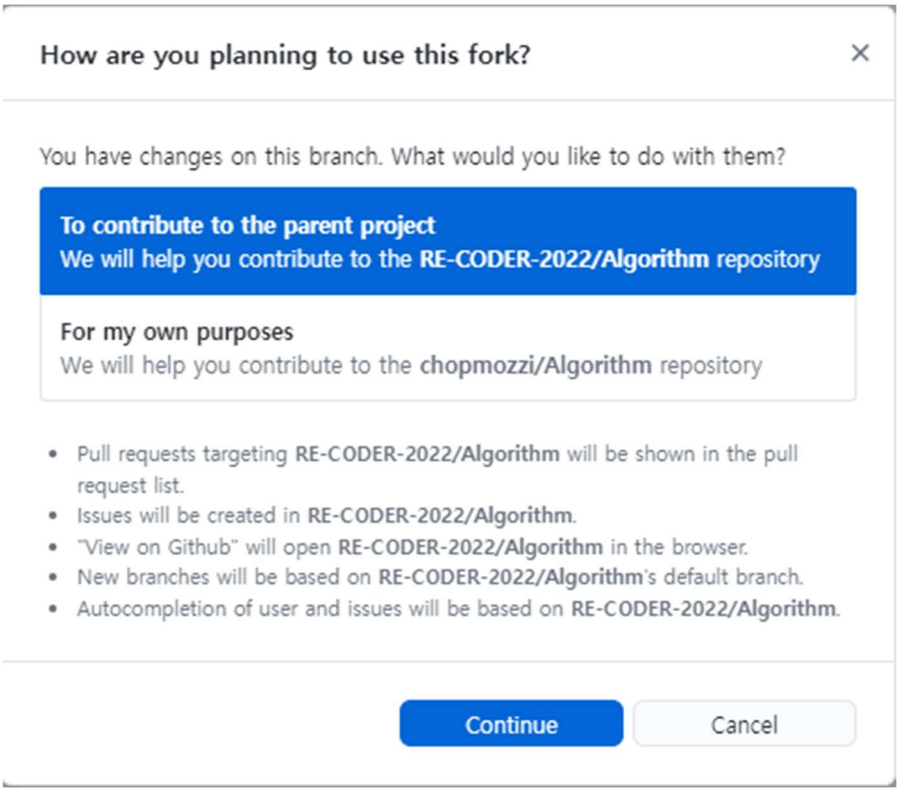



그 후, Github Desktop에서 File -> Clone a repository를 누른 뒤 URL에서 URL을 복사해줍시다.

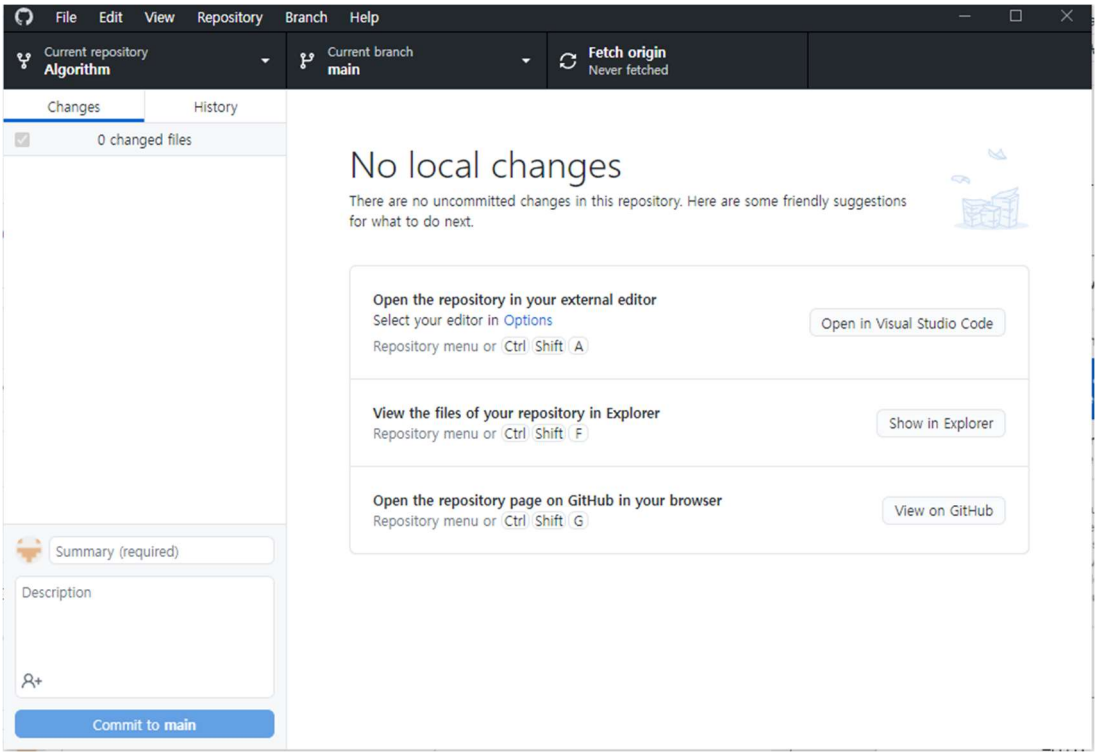



위와 같은 방식으로 하시면 됩니다.

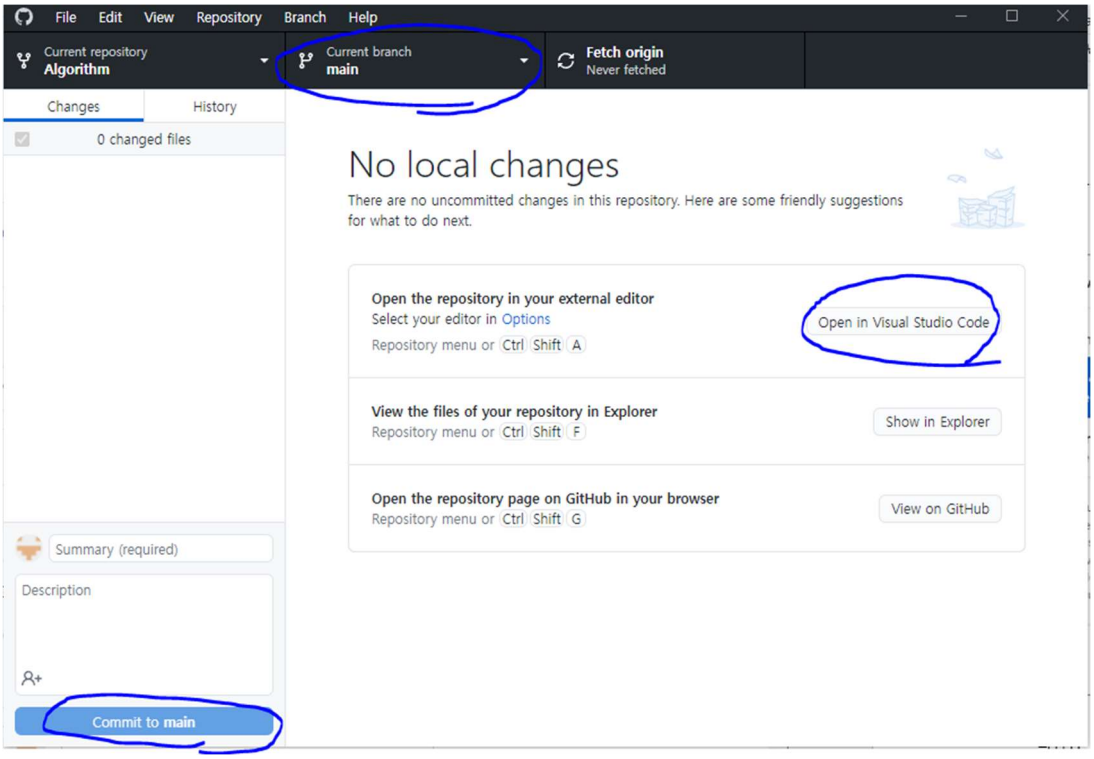
아니면 GitHub.com에서 Repository눌러서 가져오셔도 됩니다.




그러면 위와 같이 창이 뜨는데 우리는 RE-CODER-2022에 기록 하는게 목적이므로 위의 항목을 누르고 continue를 눌러줍시다.

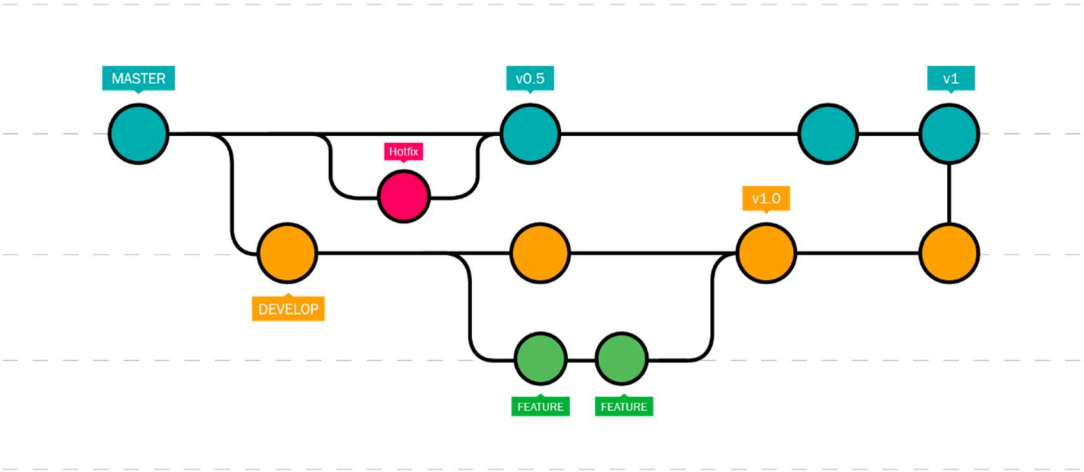



그럼 위와 같이 창이 뜹니다.




중요한 항목이 세가지 정도 있습니다. 먼저 branch가 있습니다.

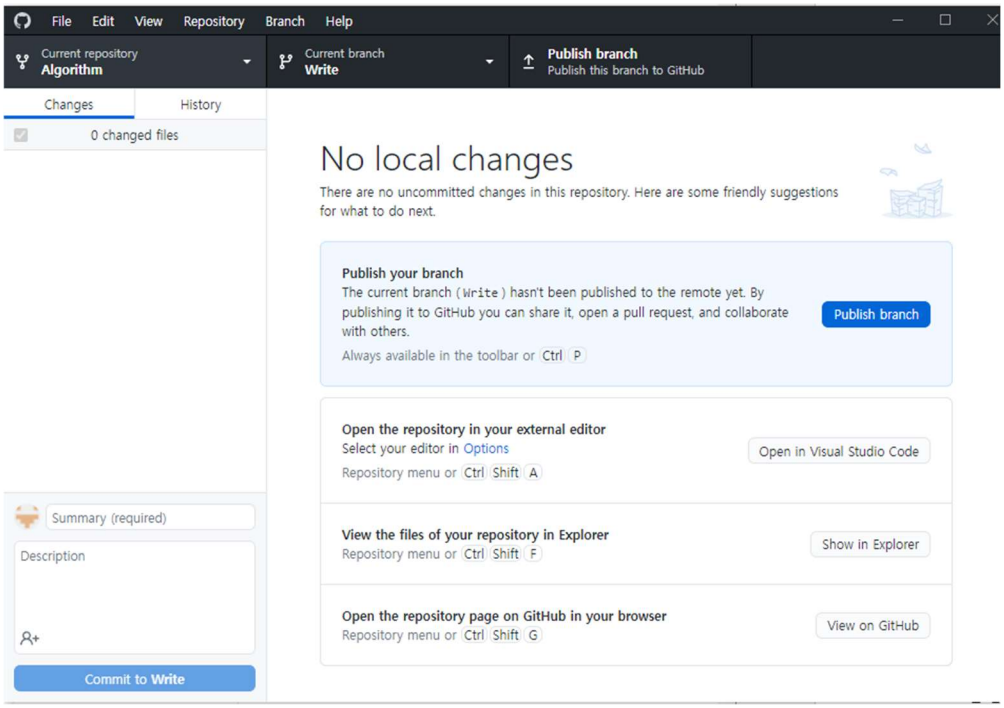
Branch는 말 그대로 프로젝트의 가지를 나타냅니다. 프로젝트의 중심인 main branch가 있습니다.




위와 같은 형식이라고 생각하시면 됩니다. MASTER를 main이라고 생각하시면 되겠습니다.

Branch를 생성하여 작업을 하고 해당 생성한 branch를 다시 main branch에 merge합니다. 이 때, 각 branch에는 수정했던 작업 기록이 남아있습니다. Maintainer는 이 작업 기록을 확인하고 main branch에 merge할지 결정합니다. 이 때, branch를 따로 생성하지 않고 main branch에서 작업할 수도 있을 것입니다. 하지만 그렇게 될 경우 Pull Request를 날릴 때 마다 이전에 했던 작업물들도 다시 Pull Request가 됩니다. Main branch의 작업 기록이

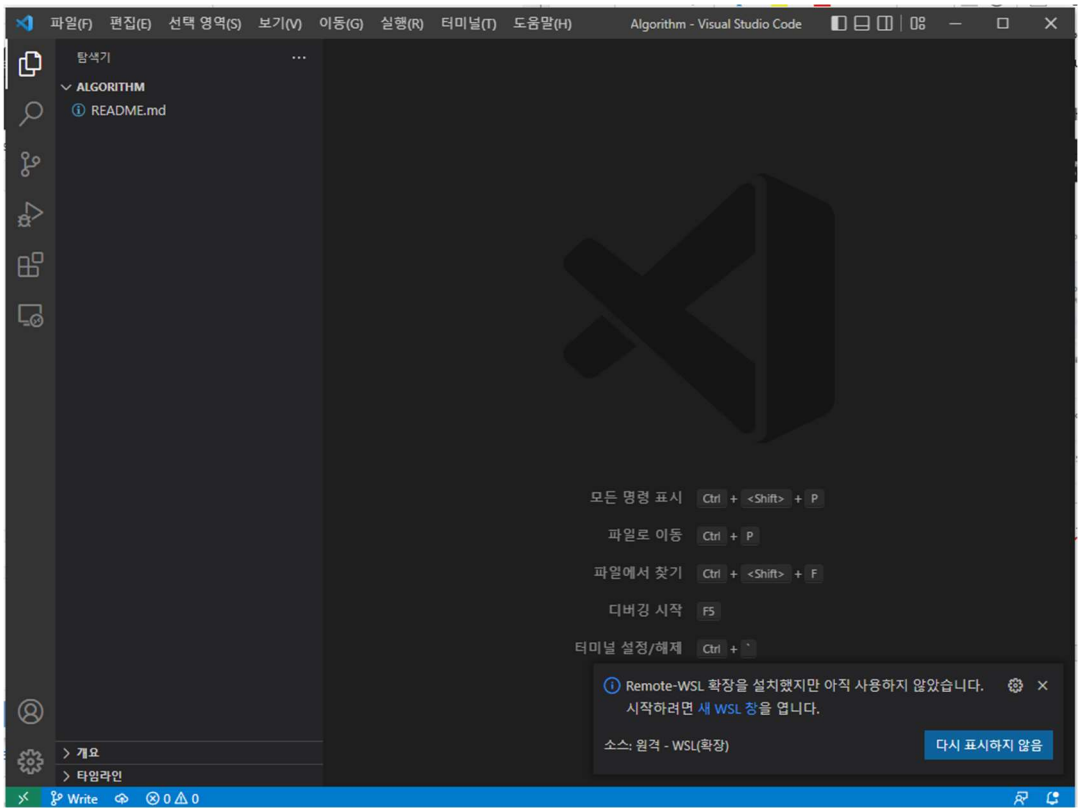
남아있기 때문입니다. 따라서 알고리즘 공부한 내용을 작성하고 commit 후 Pull Request 를 날리 실 때 branch를 생성했는지 꼭 확인 바랍니다.




Branch를 생성했으면 위와 같은 화면이 되어있을 것입니다. Branch의 항목이 자신이 만든

branch인지 확인할 수 있도록 합니다.

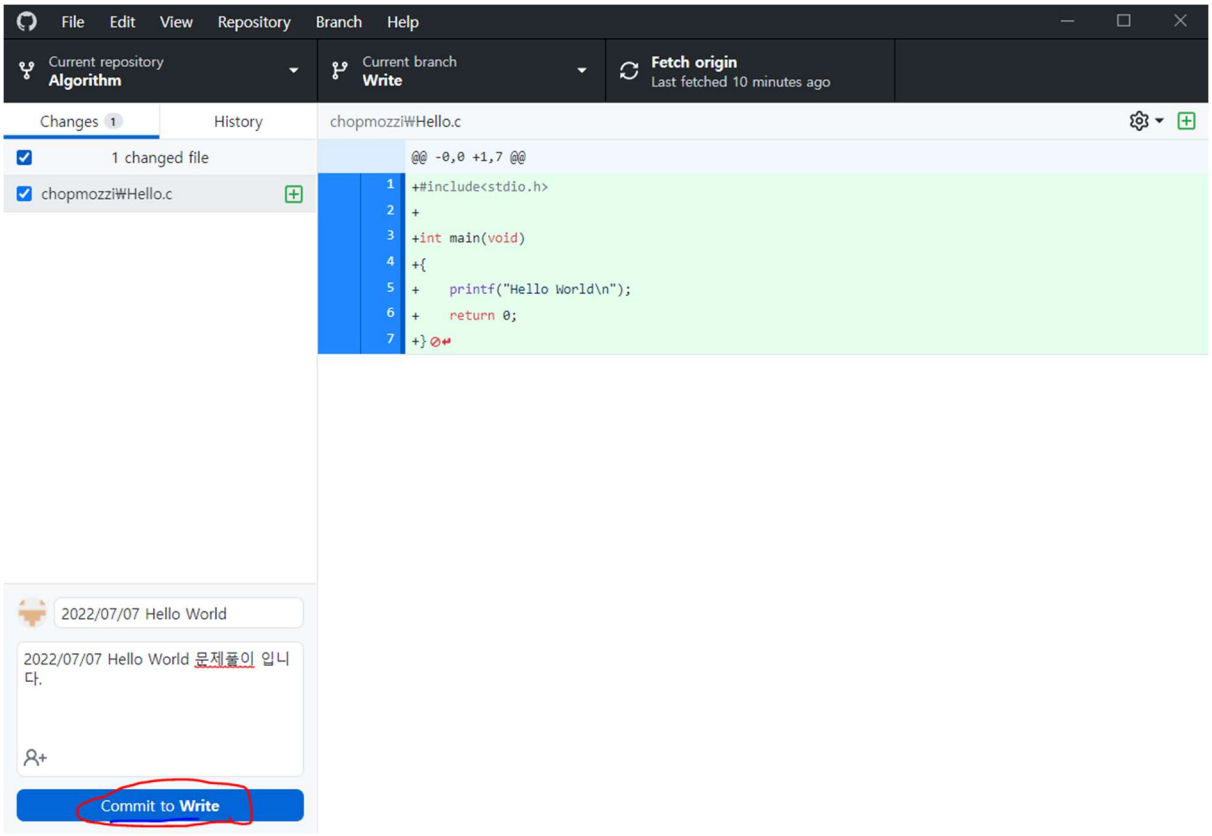
그 후, Open in Visual Studio Code를 이용하여 vscode를 열어주도록 합시다.



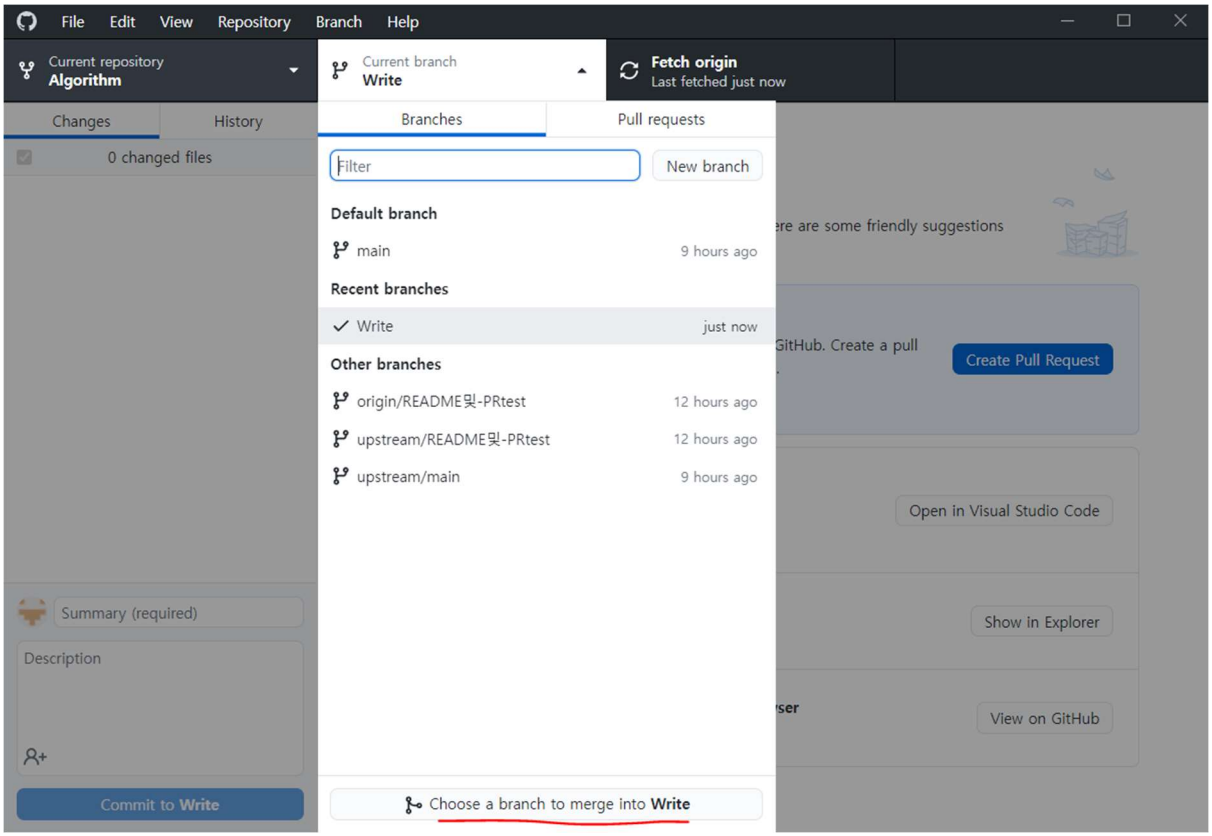

그러면 위와 같이 창이 뜰 것입니다.

그러면 탐색기에서 우클릭 -> 새폴더를 하여 본인 ID, 혹은 닉네임, 실명으로 자기 자신의 폴더를 만들어줍니다.

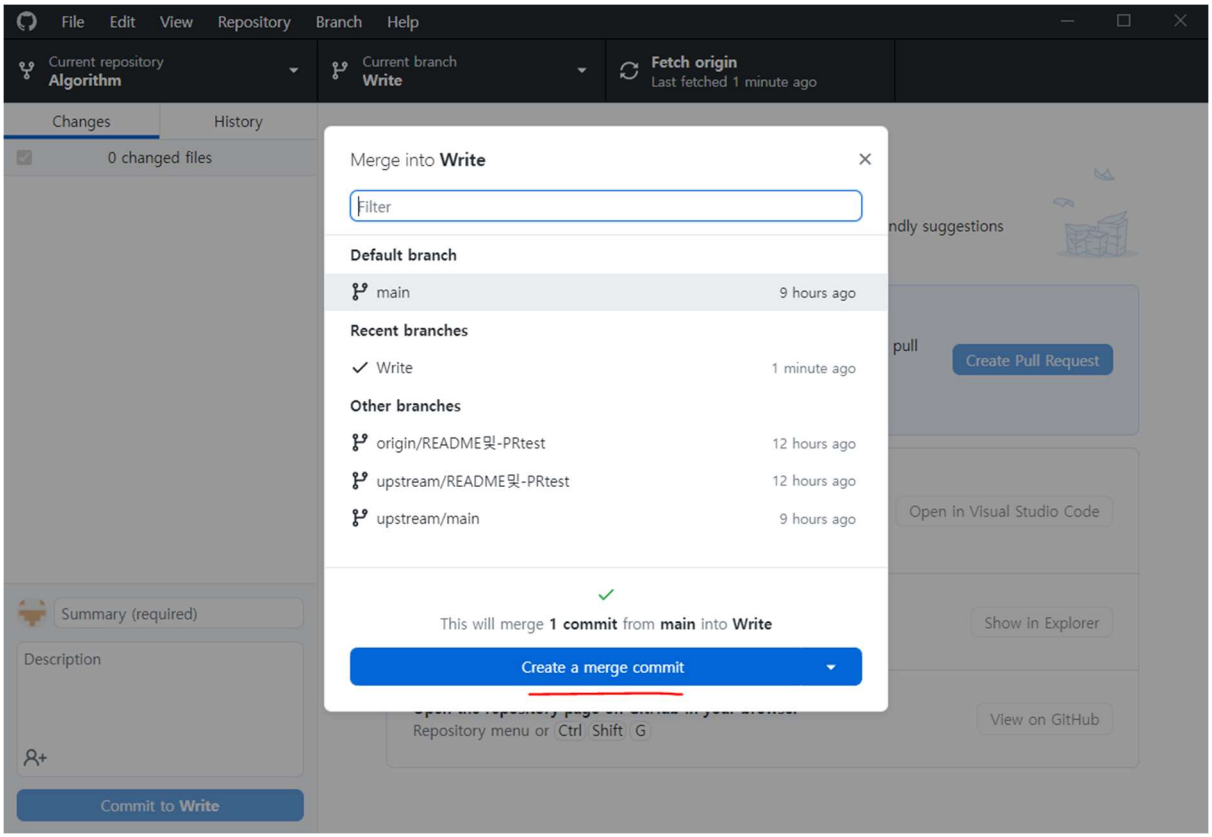

작업을 한 모습이다.



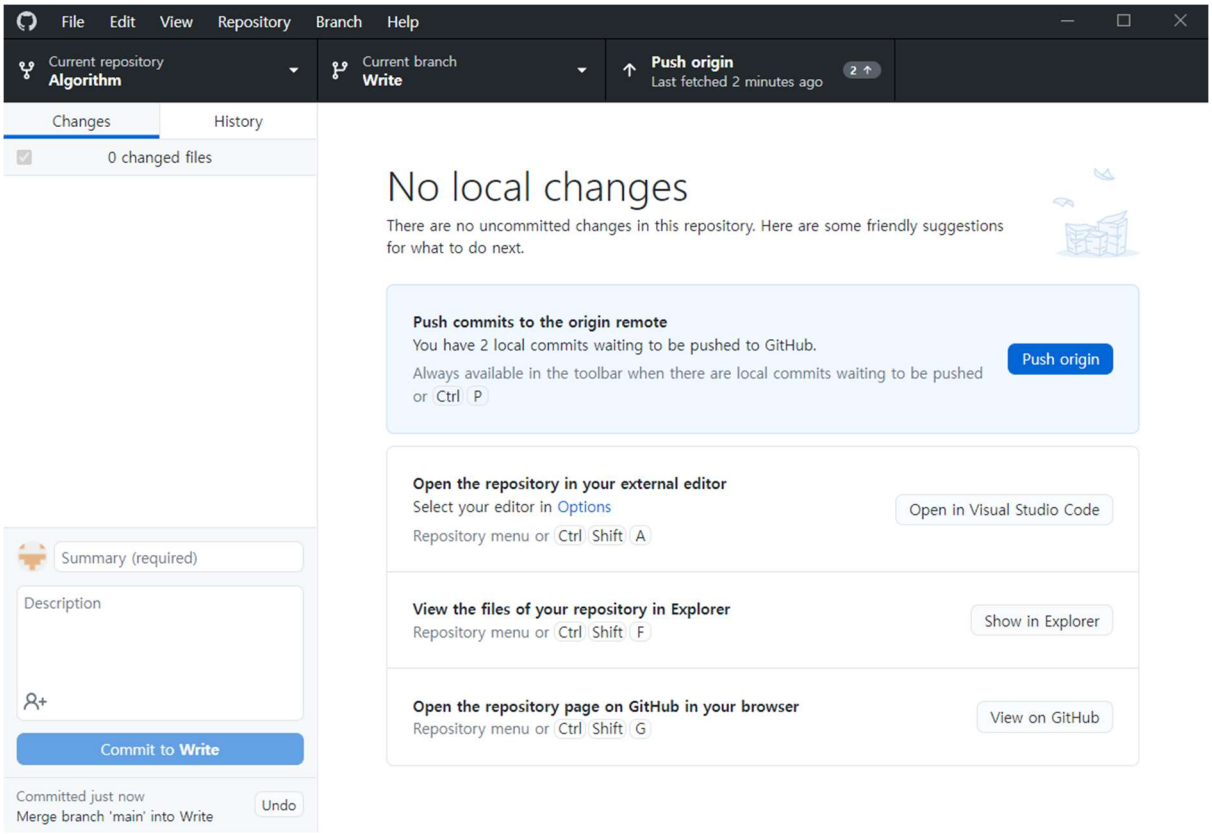

Commit 내용을 작성하고 commit을 해줍니다.



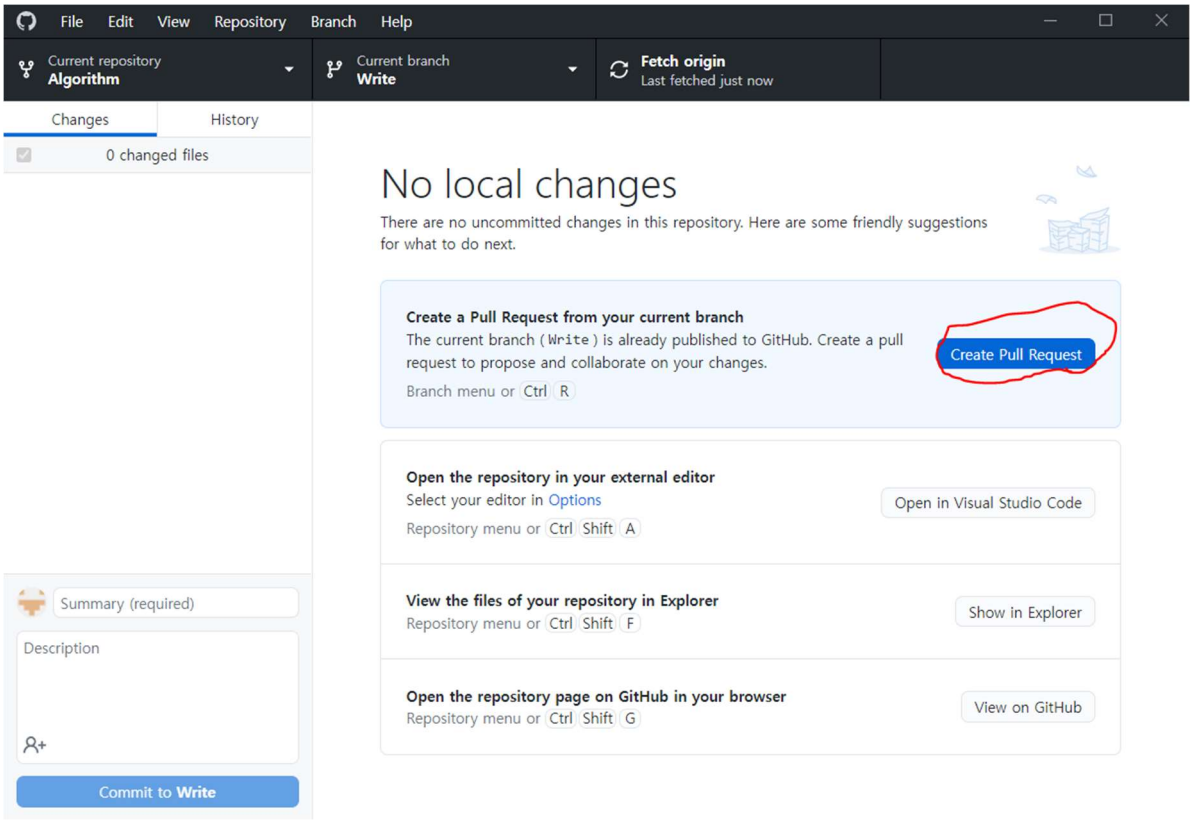

생성한 branch를 main branch에 병합해 줍니다.



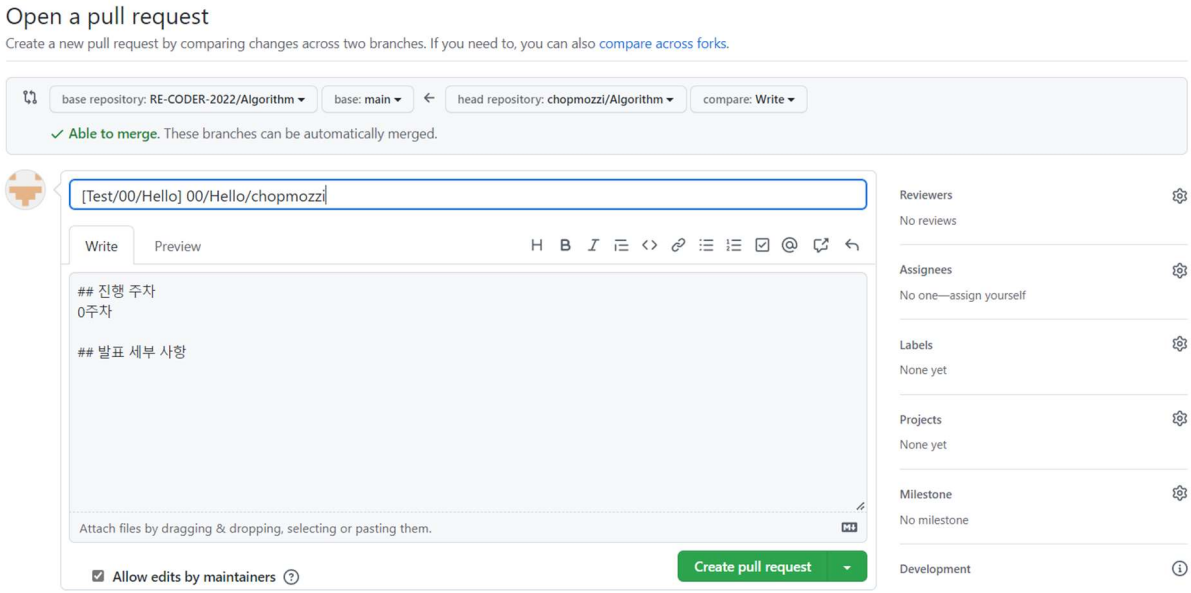

Main branch에 merge해줍니다.




Push origin버튼을 눌러 github repository에 push 해줍니다.




Create Pull Request 버튼을 눌러 Pull Request를 해줍니다.

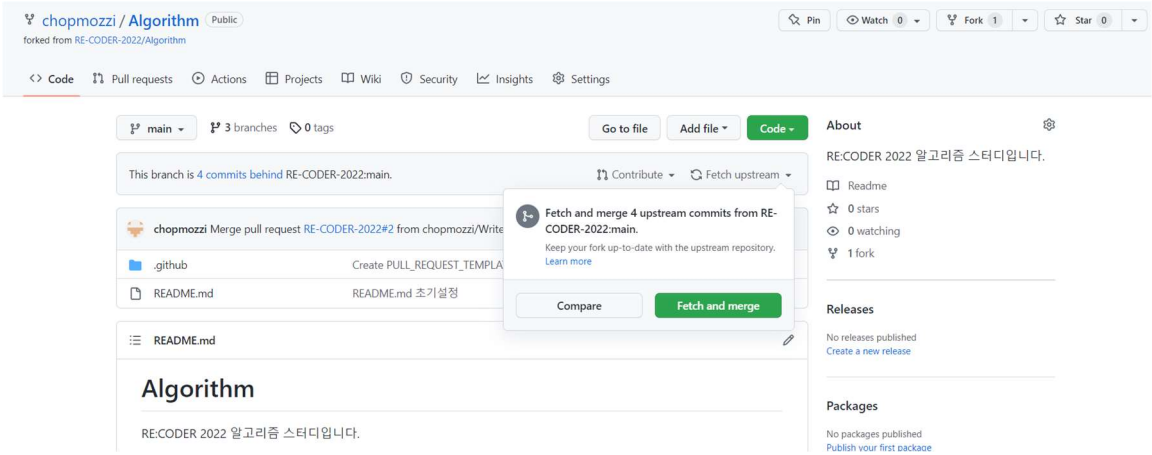



양식에 맞게 Pull Request를 작성해줍니다.

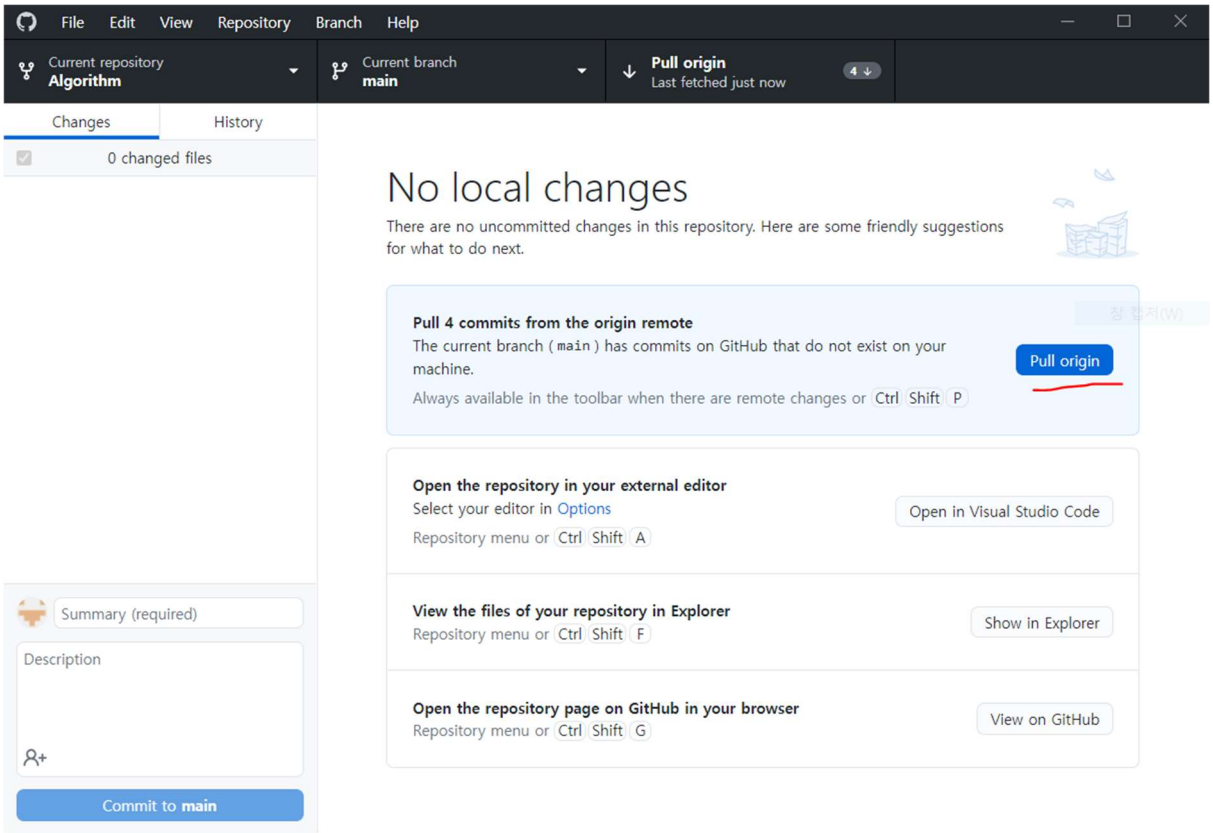
이러한 방식으로 Github Desktop을 이용하여 Pull Request를 진행할 수 있습니다.

Pull Request가 받아들여져 Merge가 되면 기존의 Repository(여기서는 RE:CODER의 Repository)의 내용이 변경이 됩니다. 그러나 fork한 repository(자신의 repository)에는 해당 내용이 업데이트되지 않은 상황입니다. 따라서 fork 한 repository의 내용을 최신화를 해주어

야 합니다.




Fork를 한 repository에 가보면 위에 Fetch upstream이라고 해서 항목이 있습니다. 위의 내용을 눌러 Fetch and merge를 통해 fork한 repository의 내용을 업데이트해줍니다.



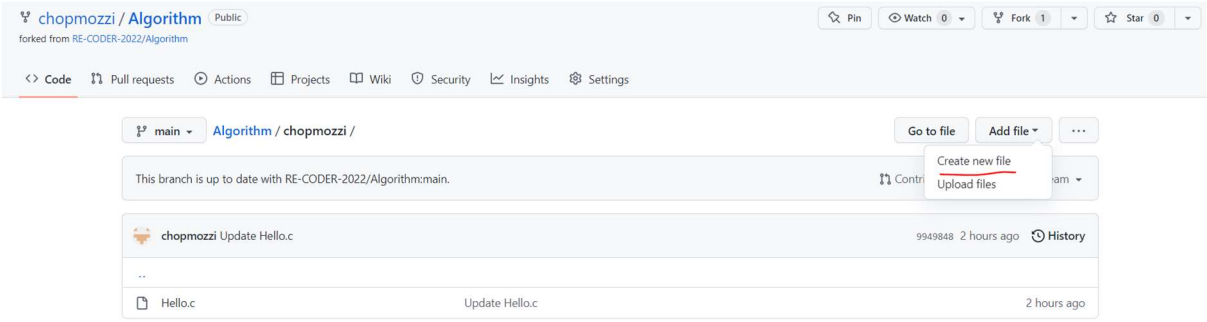

그럼 Github Desktop에도 해당 과 같은 항목이 나옵니다. Pull origin버튼을 통해 pull을 하여

로컬 저장소에도 업데이트를 해주시면 되겠습니다.

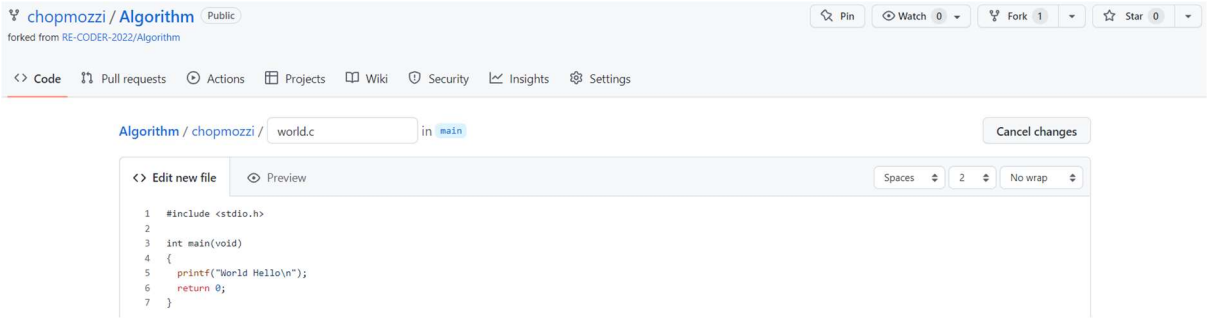
2. 깃허브 홈페이지 내에서 직접 수정

   가장 간단한 방법입니다.

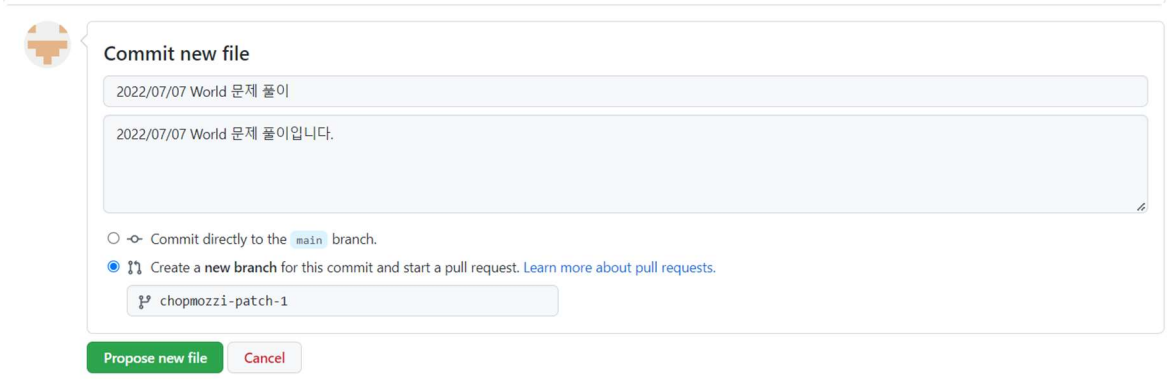
   그러나 깃의 사용법을 익히기에는 부적절하여 사용을 그렇게 추천드리지는 않습니다.




   깃허브 내의 Add file을 이용해서 파일을 추가할 수 있습니다.




   해당 파일을 작성하고




   Create a new branch for this commit and start a pull request를 이용해 PR을 할 수 있습니다.

   위와 같은 방식들을 Pull Request를 할 수 있습니다.

   이번 알고리즘 스터디에서는 해당 Pull Request를 이용해 자기가 발표할 문제 코드, 내용 등을 기록할 예정입니다.