

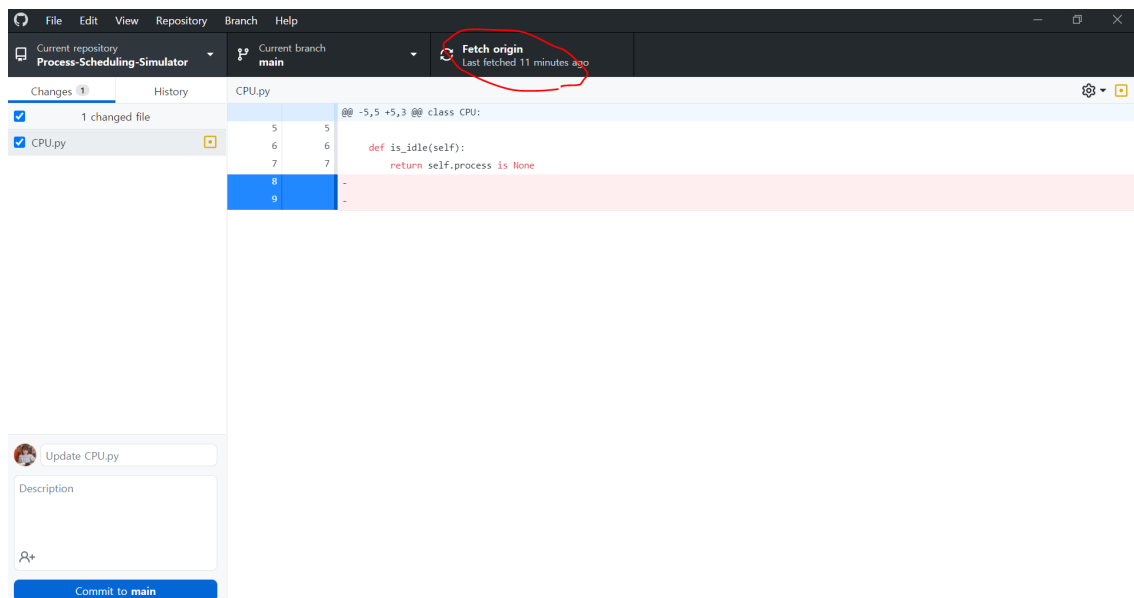
[Github] Forking Workflow

Github에서 프로젝트 협업하기

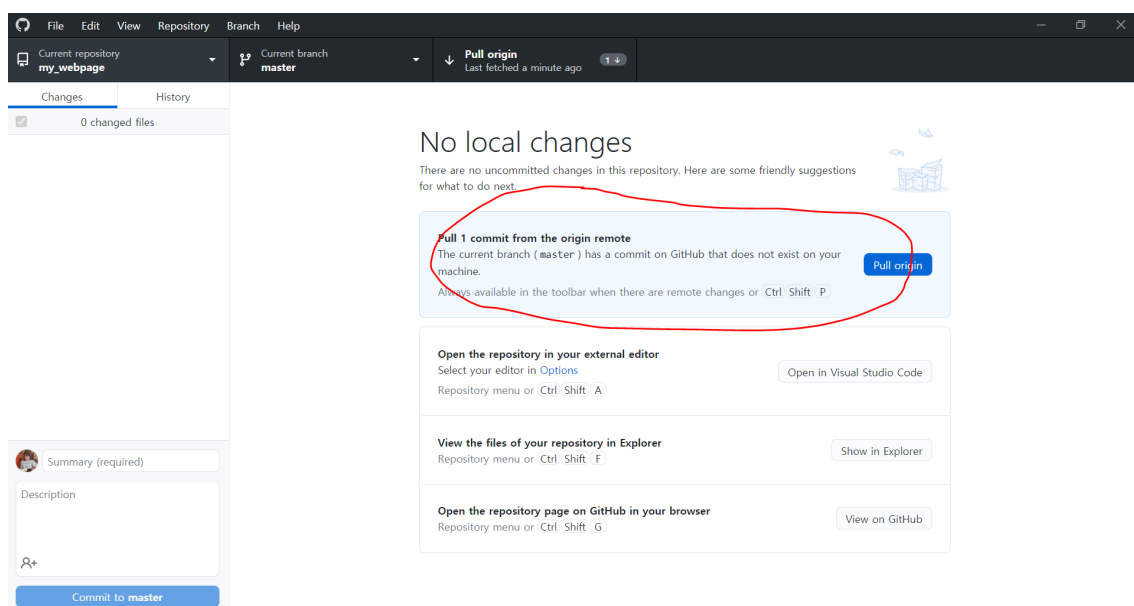
- 준비 : Github계정, [GitHub Desktop](#) 설치
- [참고 - \[Git\] FORK방식으로 협업하기](#)

용어 정리

- **원격 저장소** : 파일이 원격 저장소 전용 서버에서 관리되며 여러 사람이 함께 공유하기 위한 저장소 (쉽게 말해서 팀장이 관리하는 github 사이트)
- **로컬 저장소** : 내 PC에 파일이 저장되는 개인 전용 저장소(쉽게 말해서 개인 PC)
- **fetch origin** : 원격 저장소의 최신 이력을 확인할 수 있음

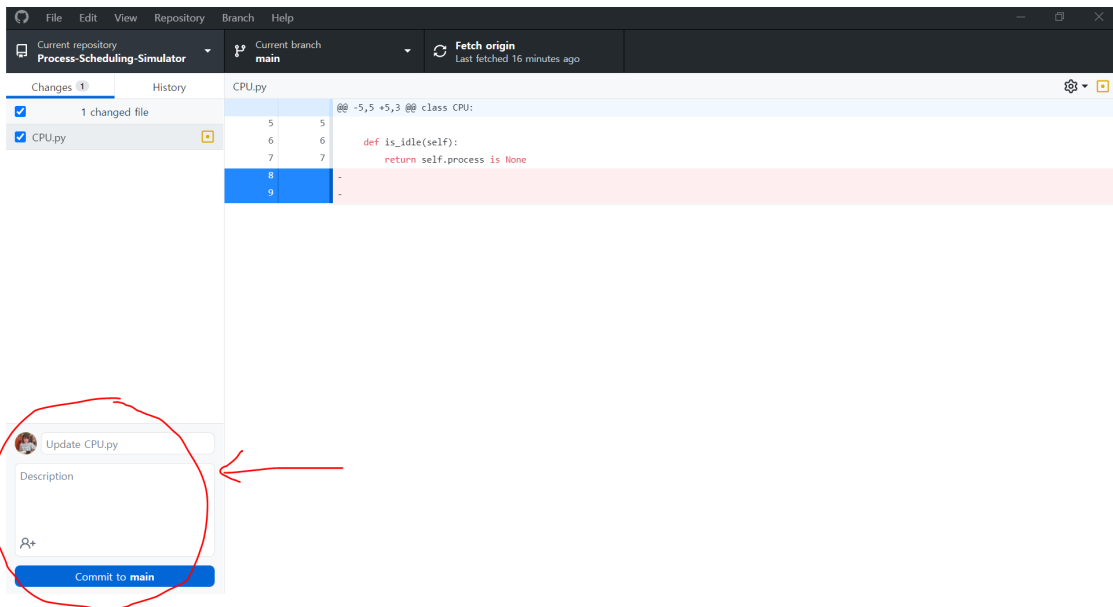


- **pull** : 원격 저장소의 최신 이력을 확인한 후 변경사항이 있으면 내 로컬저장소에 반영되어 변경됨



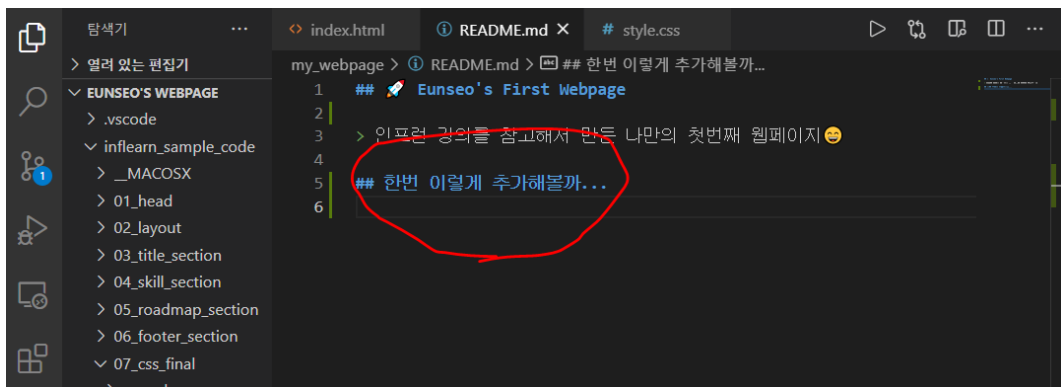
- **commit** : 파일 및 폴더의 추가/변경 사항들에 대해 기록을 하는 것. 내가 작성/수정한 코드에 대한 설명을 적어 과거 변경 이력을 저장하고 다른 팀원들에게도 공유할 수 있다.

[Git commit 메시지 규칙](#) → 여기 참고하시면 돼요. 되도록 규칙 지켜서 커밋 작성해봐요!!

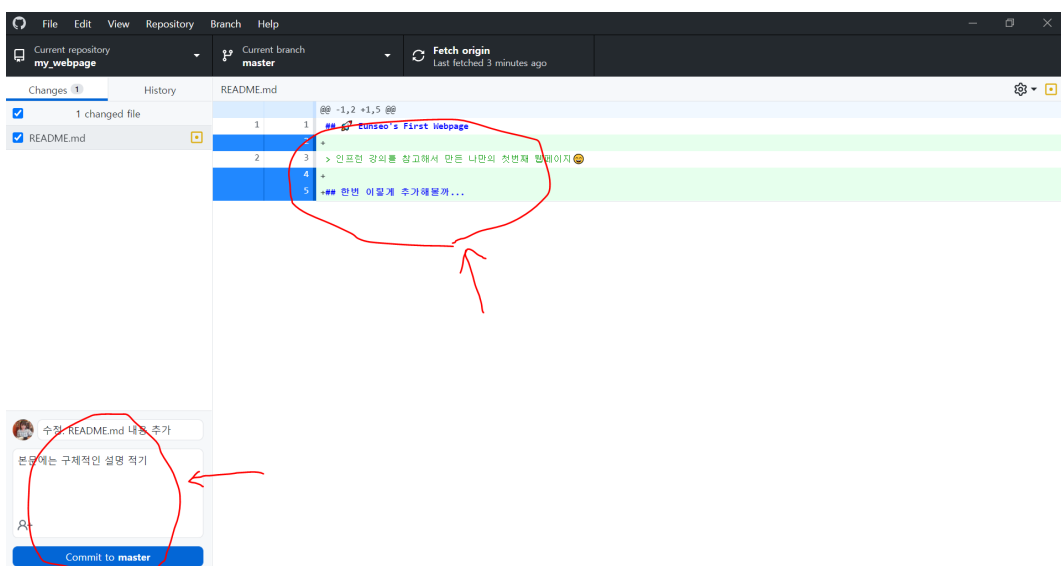


- **push** : 로컬저장소의 변경사항을 원격저장소로 반영

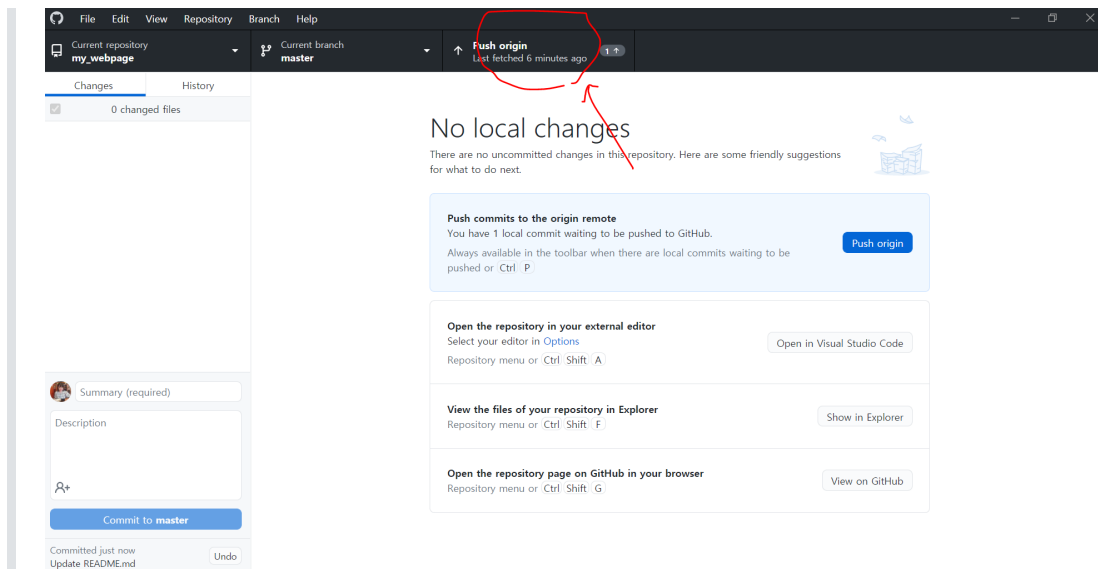
◦ 예를 들어 이렇게 원격저장소와 연결된 로컬저장소에서 변경 사항이 있으면



- 이렇게 변경사항이 Github Desktop에 반영됨을 확인할 수 있다.
- 변경사항을 원격저장소에 push하려면 왼쪽 하단에 Commit 을 작성한 후

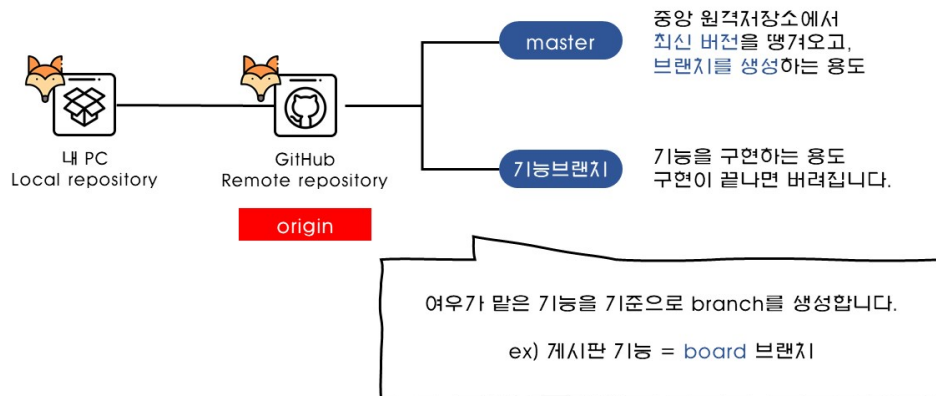


- **push** 를 통해 원격저장소에 반영한다.

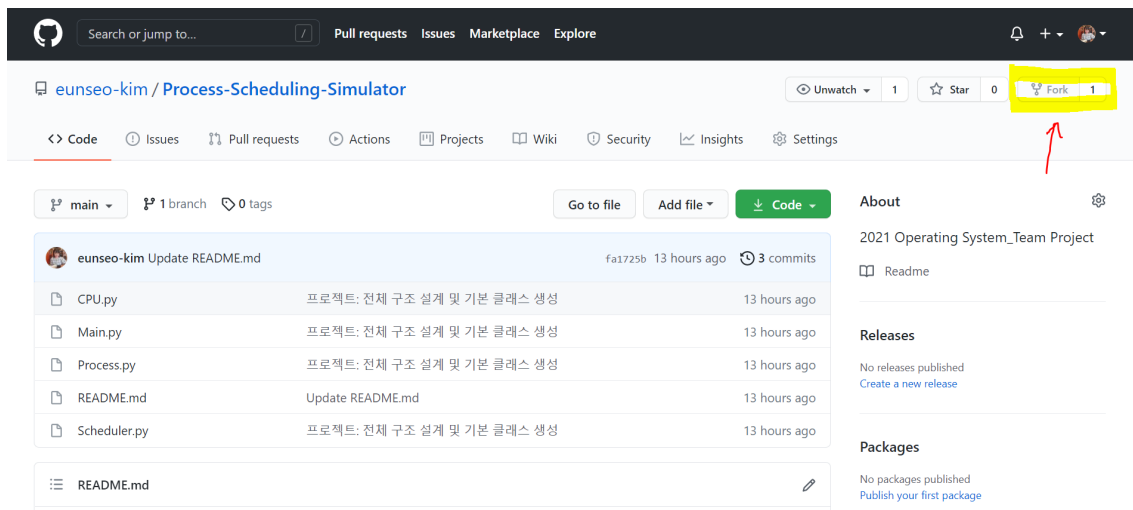


- **branch** : 브랜치란 프로젝트 내부에서 독립적인 공간을 새롭게 만드는 것. 기본적으로 master(main) branch는 존재한다.
 - **master(main) branch** : master branch에는 완성된 코드만 저장해서 관리한다.
 - **기능 branch** : 각 팀원이 특정 기능을 구현할 때 **기능 branch**를 생성해서 그 안에서 코드를 구현하도록 한다.
 - **master** 브랜치에는 **기능 branch**의 변경사항이 반영되지 않는다.
 - **기능 branch**가 완성되면 **master** 브랜치에 내가 만든 **기능 branch** 코드를 합칠 수 있다. 최종적으로 코드가 병합되면 **기능 branch**는 삭제한다.

3. 각 브랜치의 역할을 알아봅시다.

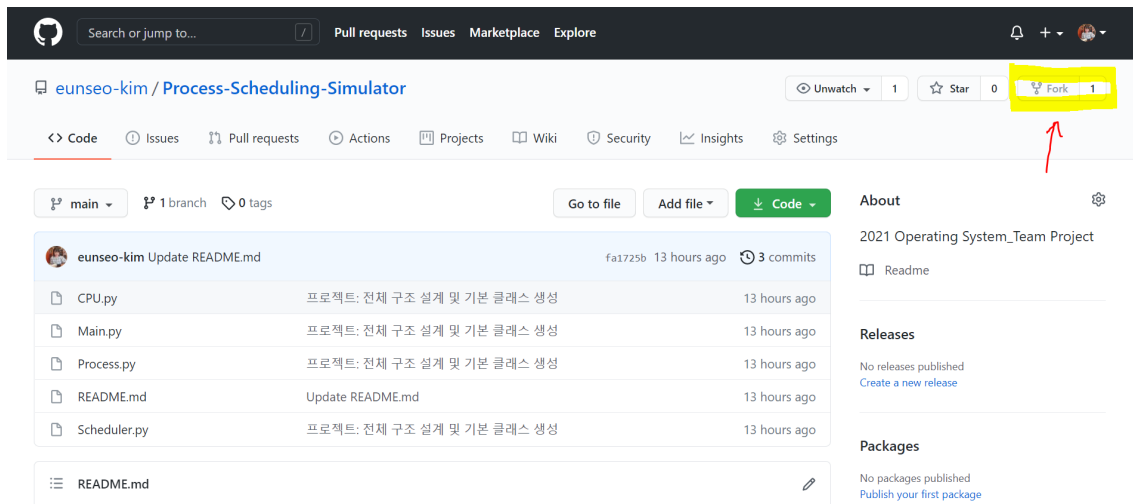


- **fork** : 다른 사람의 repository를 나의 개인 Github의 repository로 그대로 복제하는 기능

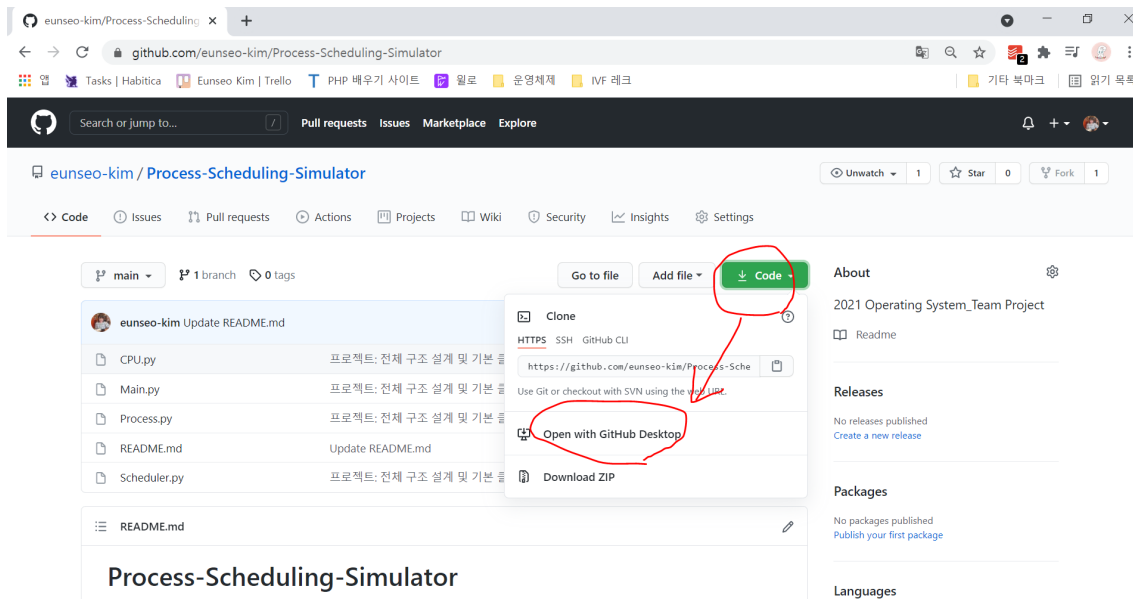


Forking Workflow 방법

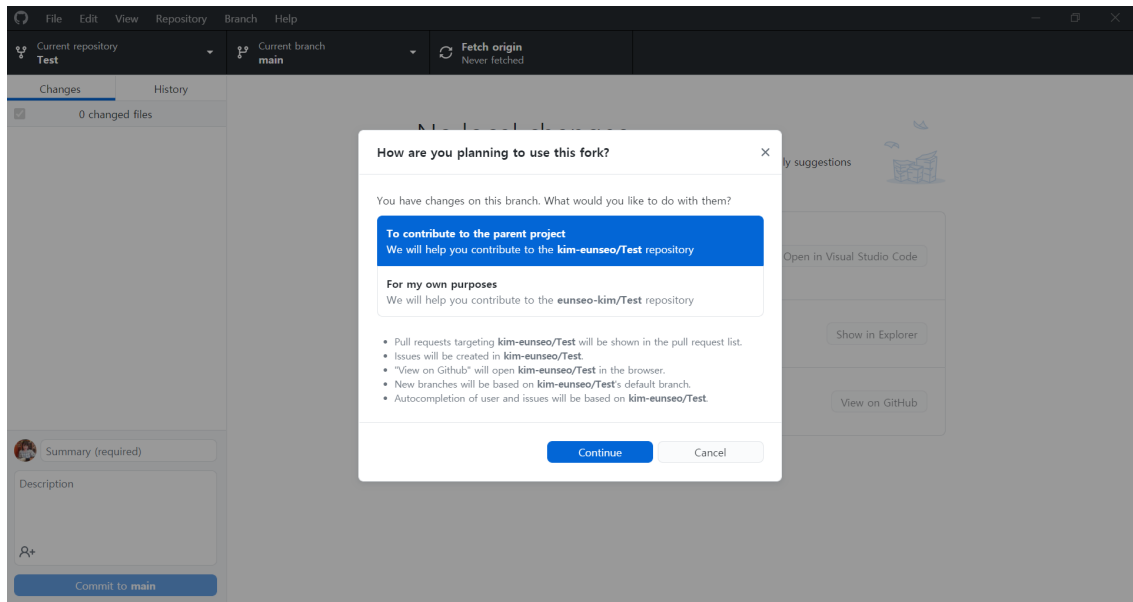
1. 팀장의 github repository를 나의 github으로 Fork 한다.



2. 자신의 github 들어가서 Clone 한다.



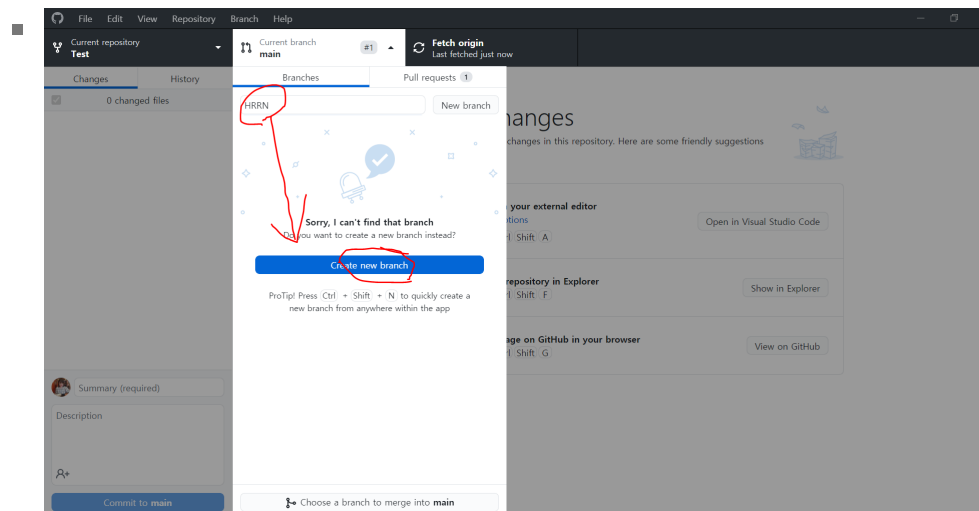
3. Local Path 설정하고 위에게 누르기(팀장의 원격저장소에 기여하도록 설정)



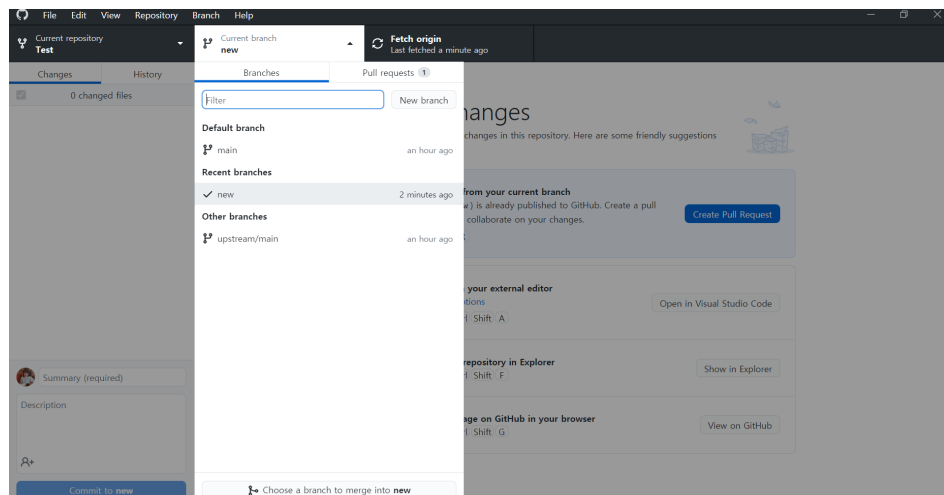
4. 팀원은 코드를 작성하고 commit, push를 하면서 개인(팀원) 원격저장소에서 코드를 완성시킨다.

내 로컬저장소에서 코드 작업하는 과정

1. 팀원은 자신의 로컬저장소에 **기능 branch**를 생성한다. 기능 branch의 이름은 내가 구현하고자 하는 기능을 이름으로 짓는다.



- 이렇게 branch를 만든 후에, current branch를 만든 브랜치로 바꾸고 작업해주세요.

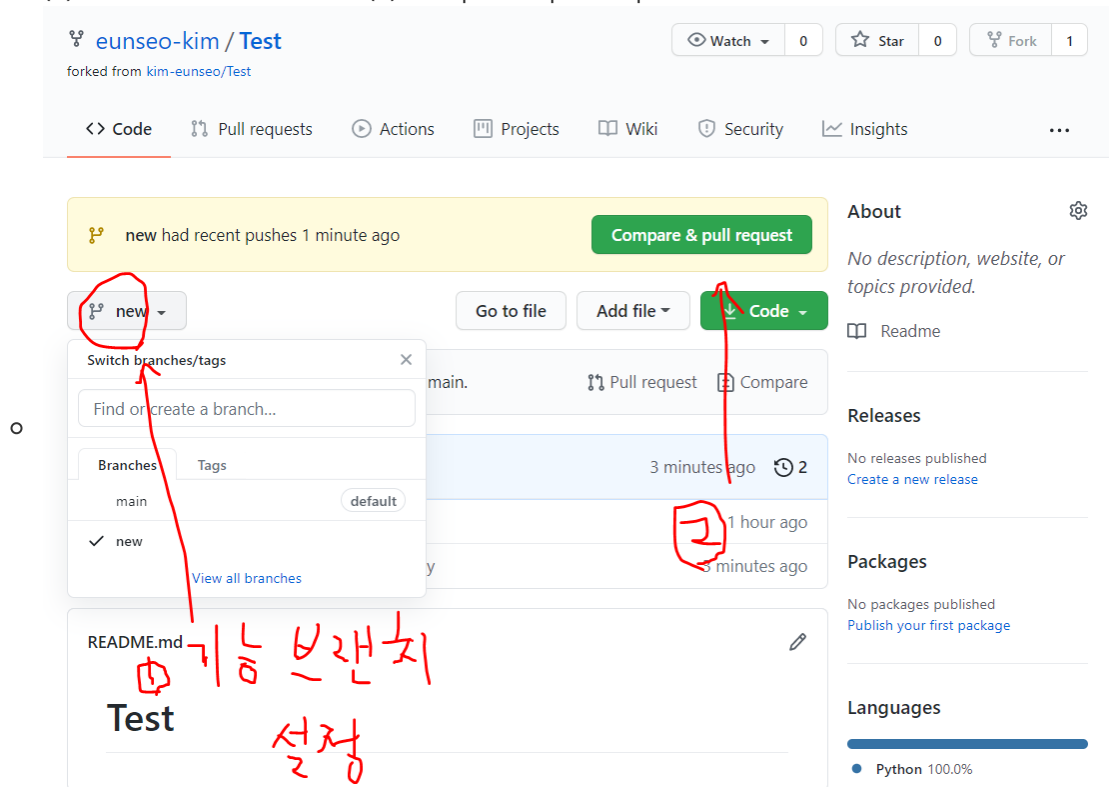


2. 팀원은 자신이 만든 기능 branch 안에서 작업을 한다.
3. 다 구현하면 팀원의 기능 branch를 팀장의 main branch로 merge(병합) 해달라고 승인 요청을 한다. (5번 과정 참고)
4. merge 승인이 완료되면 기능 branch를 삭제한다.
 - 이 과정을 계속 반복.
 - 중요한 것은 작업 시작 전에 항상 팀장 github에서 최신화된 버전을 팀원(나의)의 로컬 저장소로 가져와야 된다! (fetch origin → pull 하기)

5. 팀원이 코드를 다 만들면 팀원의 깃허브에서 pull request를 누르면 팀장에게 승인 요청을 보낼 수 있다.

즉, 나의 기능 branch를 팀장의 main branch로 merge(병합) 해달라고 승인 요청을 보낸다.

- (1) 기능 브랜치를 누르고 → (2) Compare & pull request를 누른다.



- 아래와 같이 팀장 레포지토리와 main 브랜치, 팀원 레포지토리와 기능 브랜치를 확인하고 제목, comment에 추가 설명 작성한 뒤에 Create pull request를 누르면 승인 요청이 간다.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base repository: kim-eunseo/Test base: main head repository: eunseo-kim/Test compare: new

✓ Able to merge. These branches can be automatically merged.

Create eeeee.py

Write Preview

Helpful resources
GitHub Community Guidelines

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

☒ Allow edits by maintainers ?

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

1 commit 1 file changed 0 comments 1 contributor

Commits on Apr 05, 2021

Create eeeee.py 88ea1cb

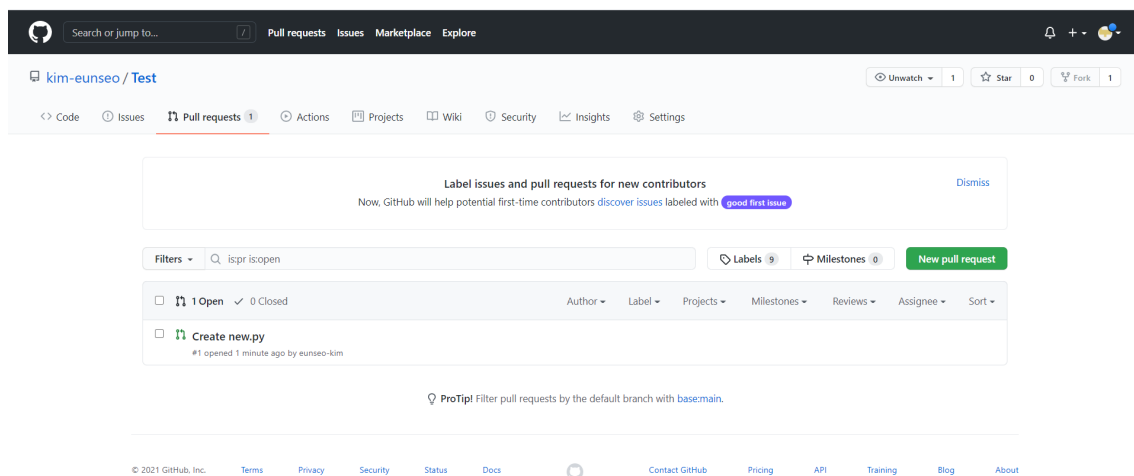
Showing 1 changed file with 1 addition and 0 deletions.

Unified Split

1 eeeee.py

```
... @@ -0,0 +1 @@  
1 + print("eeeeeeeeee")
```

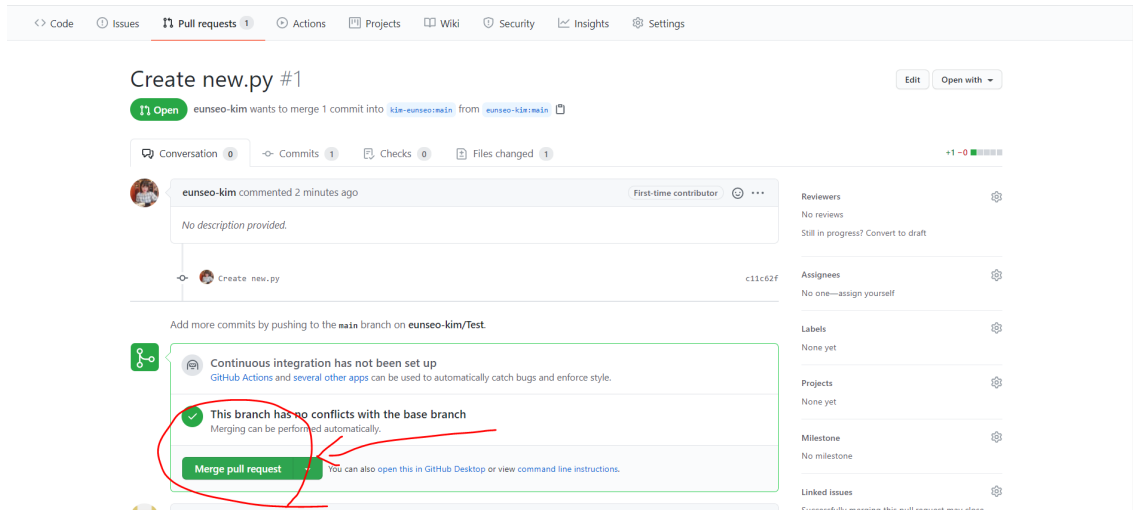
6. 팀장의 github에 들어가면 이렇게 팀원이 pull request를 보낸 것을 확인할 수 있다.



7. 코드 검토 후 이상이 없으면 팀장이 `Merge pull request`를 해서 요청을 승인한다. 이렇게 되면 팀원이 작성한 코드가 전체 코드에 적용이 된다!

- 이때, merge하게 되면 팀장의 원격저장소에 `기능 branch`가 추가되는 것이 아니라, `main branch`의 코드 자체가 변하는 것이다!

■ main branch + 팀원의 기능 branch = main branch

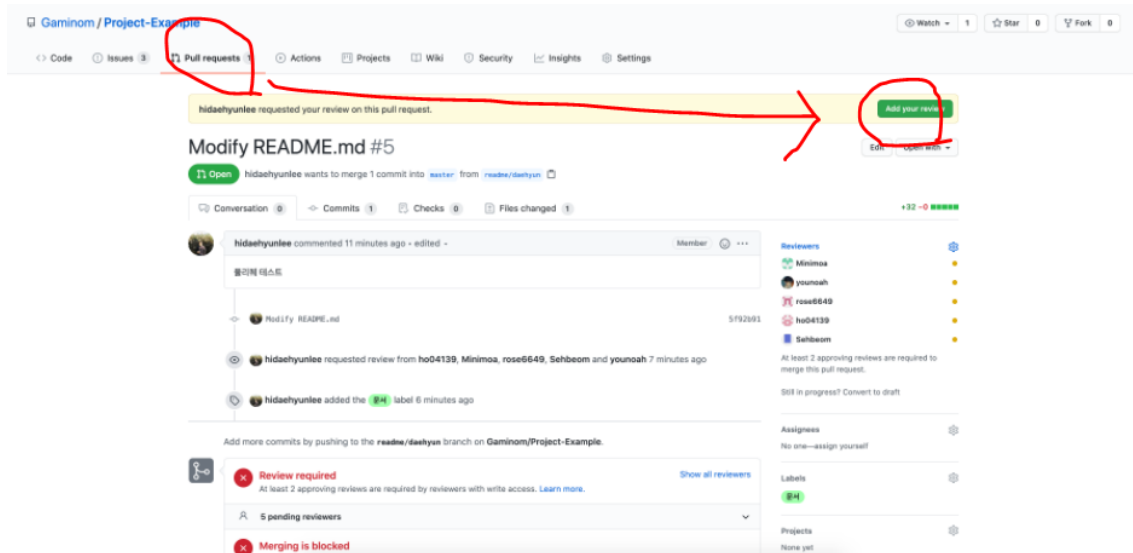


병합 관련해서는 이 글을 읽어보세요. 쉽게 설명되어 있음 → (<https://mintparc.tistory.com/141?category=835965>)

코드 리뷰 방법

- [코드리뷰 방법 참고 블로그](#)

1. Pull Request에 들어가서 Add your review 를 클릭한다.



2. (필요하다면)코멘트를 적고 해당 Pull Request에 대하여 리뷰한다.

- **Comment** : 그냥 코멘트만 달아준다.
- **Approve** : 이 코드가 merge(main branch에 병합되는 것)되는 것에 동의한다!
- **Request changes** : 코드에서 버그를 발견해서 다시 수정해달라고 요청할 수 있다.

Modify README.md #5

hidaehyunlee wants to merge 1 commit into master from readme/daehyun

Conversation 0 Commits 1 Checks 0 Files changed 1 +32 -0

Changes from all commits File filter... Jump to... 0 / 1 files viewed Review changes

32 README.md

```
@@ -1,33 @@
1 1 # Project-Example
2 +
3 + 1. **organization**을 만듭니다.**
4 +
5 + 2. **팀** 만들어서 멤버로 초대한다.**
6 +
7 + 3. **프로젝트 repository**를 만듭니다.**
8 +
9 + 4. **repository contributor**로 팀원들을 초대하고 권한을 준다.**
10 +
11 + 5. **issue, wiki** 사용법을 익힌다.**
12 +
13 + 6. **pull request** 사용법을 익힌다.**
14 +
15 + - **항상 모든 작업은 브랜치를 따서 한다. 마스터 브랜치에 직접 push 금지.**
16 +
17 + - 브랜치 이름은 `기능/사람이름` ex) `detail/daehyun`
```

Write Preview H B I

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

☒ **Comment**
Submit general feedback without explicit approval.

☐ **Approve**
Submit feedback and approve merging these changes.

☐ **Request changes**
Submit feedback that must be addressed before merging.

Submit review

3. 만약 모든 팀원들이 **Approve** 리뷰를 달았다면
Pull Request를 요청한 팀원이 직접 merge를 승인한다.
(모든 팀원이 **Approve** 하기 전에는 절대 merge하지 않기로 약속~)