

오픈소스SW기초 (2024-2)

10. Git 명령어 III (오픈소스 기여와 협업)

국립금오공과대학교

컴퓨터공학과 / 인공지능공학과

김 경 수

목차

- ① Pull request 개요
- ② Pull request 실습

학습 목표

- ① Fork와 pull request 작업의 필요성과 역할에 대해 이해하고 이를 프로젝트 기여에 활용하는 방법을 설명할 수 있다.
- ② Git을 이용하여 오픈소스에 기여(contribution)하는 방법을 이해하고 pull request를 활용하여 간단한 프로젝트 기여 작업을 수행할 수 있다.
- ③ 조별 활동을 통해 git을 활용한 오픈소스 기여 실습을 진행할 수 있다.

Pull request 개요

프로젝트 협업의 또 다른 방법

- 프로젝트 협력자(collaborator)

- 프로젝트의 공동 책임자로, 프로젝트의 관리자(소유자)가 직접 지정
- 해당 원격 저장소에 대한 **commit**, **push**, **pull** 등의 작업을 자유롭게 수행할 수 있음

- 프로젝트 기여자(contributor)

- 현재 프로젝트에 **commit**하는 모든 사용자들을 통칭
- 협력자와는 달리 해당 프로젝트의 원격 저장소에 **push** 명령을 수행할 권한은 없음.
- 대신 **pull request** 기능을 사용하여 자신이 **commit**한 내용을 프로젝트에 최종적으로 반영할 것인가를 프로젝트 관리자와 지속적으로 토론한 후, 관리자가 **pull request**를 최종 승인하면 **push**가 완료됨.

Pull request가 필요한 상황

- 다른 사람이 소유하고 있는 프로젝트(원격 저장소)의 소스코드를 수정하거나 그 기능을 업그레이드 하고 싶은 경우
- 이때, 해당 프로젝트에 대한 “협력자” 권한을 부여 받지 못한 경우 “기여자”로서 해당 프로젝트에 참여할 수 있다.
- 단, “기여자” 자격으로 프로젝트에 참여하는 경우, 내가 **commit**한 내용을 해당 프로젝트에 **push**하기 위해서는 관리자의 승인이 필요하다.
 - 필요 시 기여자와 관리자 사이에 지속적인 의사소통(communication)을 통해 **push** 하고자 하는 내용을 수정/보완할 수 있다.

프로젝트 기여를 위한 기여자의 pull request 절차

- ① **Fork:** 기여하려는 프로젝트의 원격 저장소를 **fork**한다.
- ② **Clone:** 자신의 계정으로 복제된(즉, ①에서 **fork**한) 원격 저장소를 로컬 저장소로 **clone**한 후 세팅 작업을 수행한다.
- ③ **Branch** 생성: 원본 프로젝트와 독립적인 작업을 위한 새로운 브랜치를 생성한다.
- ④ **Branch**에서의 작업: 새로운 브랜치에서 자유롭게 작업을 수행한다.
- ⑤ **Add, Commit, push:** 수행한 작업에 대한 **Add**와 **Commit, Push** 명령을 수행한다.
- ⑥ **Pull request** 생성: 내가 **commit** & **push**하려는 내용에 대해 관리자에게 **pull request**를 생성하여 관리자에게 전송한다.
- ⑦ **Review & merge:** 관리자가 해당 수정 사항을 리뷰하고 해당 **pull request**를 승인한 후 브랜치를 합병하면, 기여자가 작업한 내용이 프로젝트에 최종적으로 반영된다.

Fork 명령이란?

- GitHub, GitLab과 같은 클라우드 기반의 협업 플랫폼에서 원본 저장소(= 원격 저장소)를 복제하여 자신의 계정에 생성하는 일련의 작업을 의미함.
 - 다른 사람의 원격 저장소를 **fork**하면 해당 저장소가 복제되어 자신의 GitHub 계정에 원격 저장소로 생성됨.
 - 자신의 GitHub 계정으로 복제된 원격 저장소를 로컬 컴퓨터로 **clone**하면 로컬 환경에서도 작업이 가능함.
 - Fork 명령은 Git 자체에서 제공하는 명령이 아님에 유의.

Fork 명령이란?

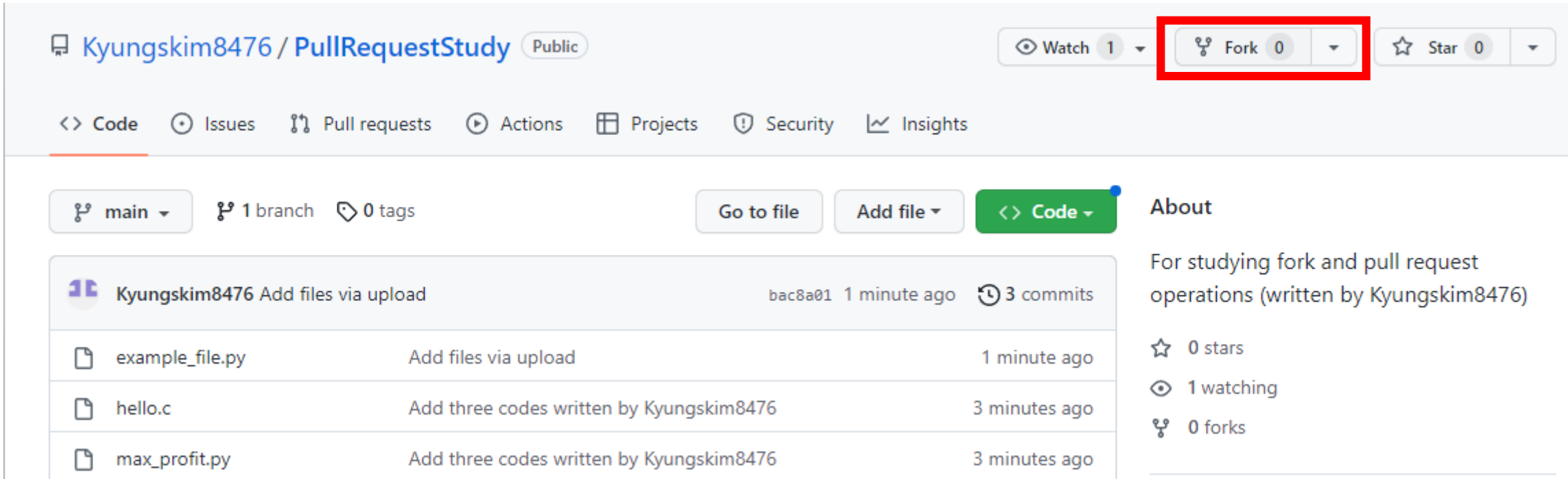
- Fork를 이용하여 타 원격 저장소(프로젝트)에 기여하는 기본 절차
 - ① 내가 기여하고자 하는 다른 사용자의 원격 저장소를 **fork**
 - ② 내 GitHub 계정에 복제된 원격 저장소를 나의 로컬 컴퓨터에 **clone**

① Fork

내가 기여하기를 원하는 원격 저장소이자
나의 pull request가 전달되는 저장소

• Target repository를 fork하는 방법

➤ Target repository의 원격 저장소에 접속한 후, 화면 오른쪽 상단의 “Fork” 버튼을 클릭하면 “Create a new fork” 기능으로 이동한다.



Kyungskim8476 / PullRequestStudy Public

Watch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

main 1 branch 0 tags

Go to file Add file Code

Kyungskim8476 Add files via upload bac8a01 1 minute ago 3 commits

example_file.py	Add files via upload	1 minute ago
hello.c	Add three codes written by Kyungskim8476	3 minutes ago
max_profit.py	Add three codes written by Kyungskim8476	3 minutes ago

About

For studying fork and pull request operations (written by Kyungskim8476)

0 stars

1 watching

0 forks

① Fork


- Target repository를 fork하는 방법

- “Create a new fork”에서 저장소 이름(repository name)과 설명(description)을 입력한 후 하단의 “**Fork**” 버튼을 클릭하면 해당 원격 저장소를 **fork**할 수 있다.

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Owner *

 kyungskim-kit

Repository name *

PullRequestStudy 


By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Owner: Kyungskim8476, Contributor: kyungskim-kit

☒ Copy the `main` branch only

Contribute back to Kyungskim8476/PullRequestStudy by adding your own branch. [Learn more.](#)

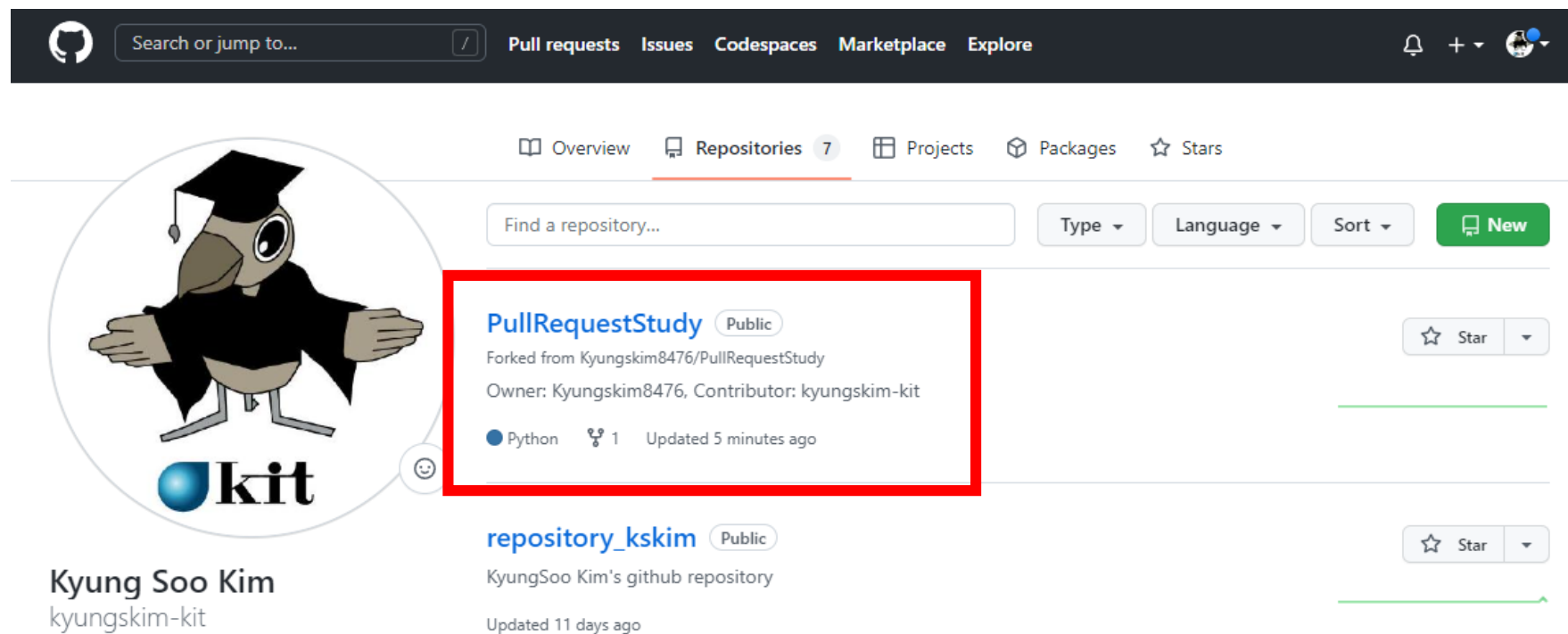
 You are creating a fork in your personal account.

Create fork

① Fork

- Target repository에 대해 **fork**가 완료되면 나의 계정에 **fork**한 저장소가 아래의 화면과 같이 나타난다.

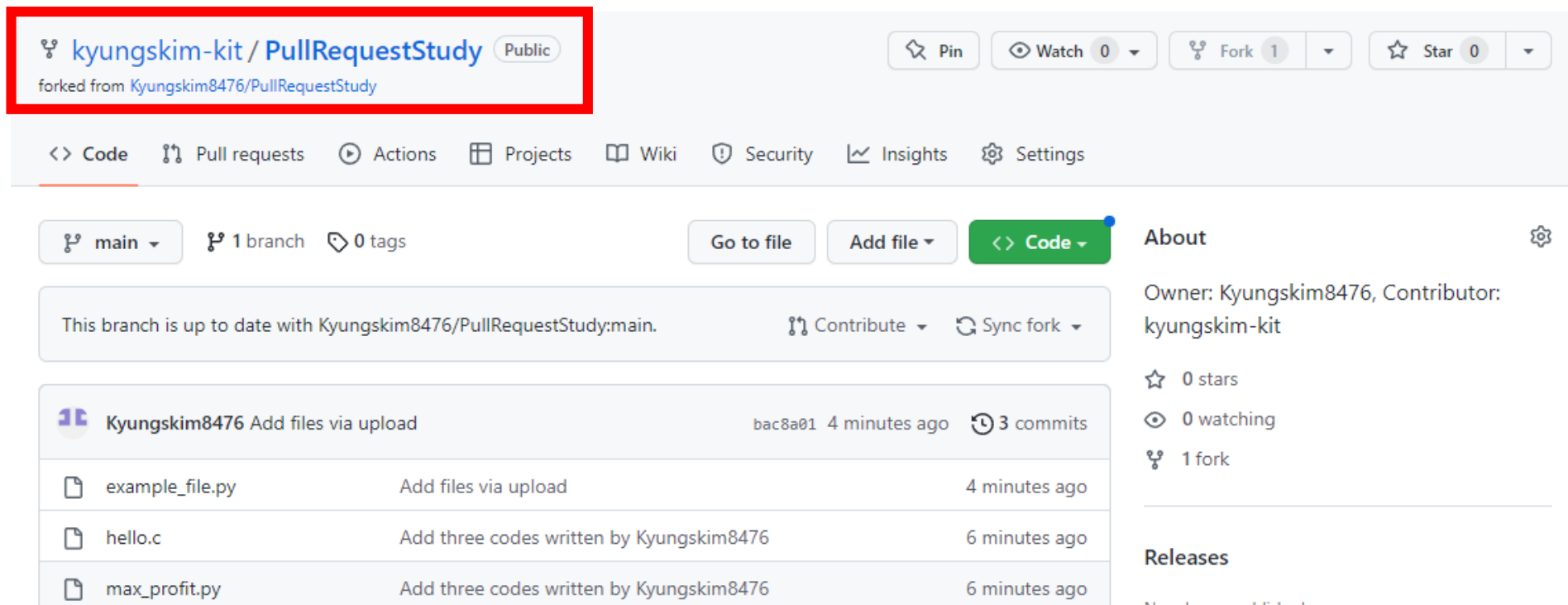
➤원격 저장소명 하단에 “*forked from original 원격 저장소 정보*”가 표기됨.



① Fork

- Target repository에 대해 **fork**가 완료되면 나의 계정에 **fork**한 저장소가 아래의 화면과 같이 나타난다.

➤원격 저장소명 하단에 “**forked from original 원격 저장소 정보**”가 표기됨.



The screenshot displays a GitHub repository page for 'kyungskim-kit / PullRequestStudy'. The repository is marked as 'Public' and is a fork of 'Kyungskim8476/PullRequestStudy'. The page shows the main branch, a list of recent commits, and the 'About' section.

Repository Information:

- Repository: kyungskim-kit / PullRequestStudy (Public)
- Forked from: Kyungskim8476/PullRequestStudy
- Buttons: Pin, Watch (0), Fork (1), Star (0)
- Navigation: Code, Pull requests, Actions, Projects, Wiki, Security, Insights, Settings
- Branches: main (1 branch), Tags: 0 tags
- Buttons: Go to file, Add file, Code

Commit History:


Commit Message	Commit Hash	Time Ago	Commits
Kyungskim8476 Add files via upload	bac8a01	4 minutes ago	3 commits
example_file.py		4 minutes ago	
hello.c		6 minutes ago	
max_profit.py		6 minutes ago	

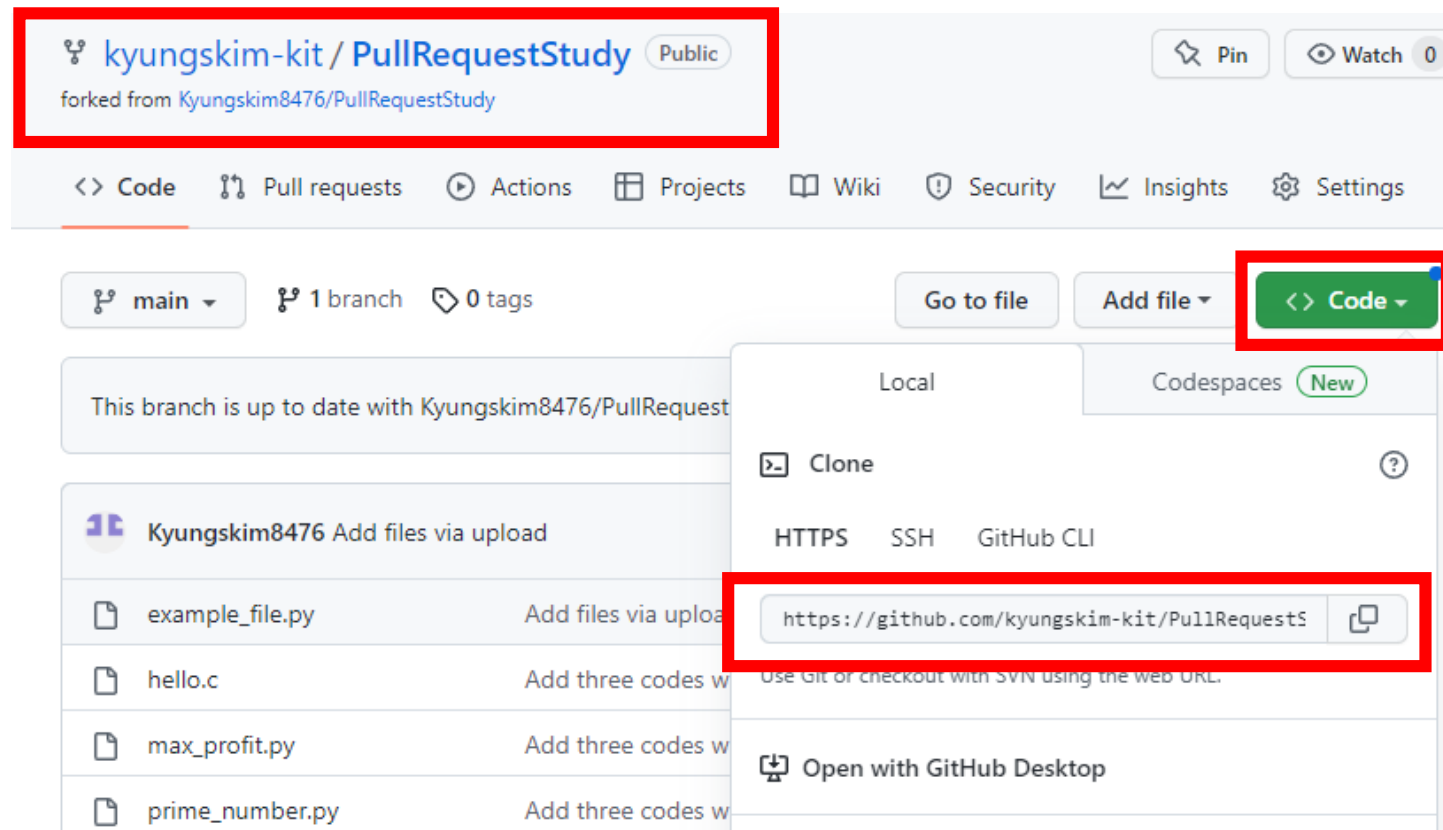
About:

- Owner: Kyungskim8476, Contributor: kyungskim-kit
- 0 stars
- 0 watching
- 1 fork

Releases:

② Clone

- 앞서 **fork**로 생성된 자신의 계정의 forked target repository에서, 화면 오른쪽 상단의 “Code” 버튼을 누른 후 “**Clone**” 창에 표시되는 URL 주소를  버튼을 눌러서 복사한다.



② Clone

- Target repository에 대한 협업을 수행하기 위해 나의 원격 저장소에 생성된 forked target repository를 **clone**한다.

git clone [방금 복사한 forked target repository의 주소]

```
KyungSooKim@DESKTOP-M68HRBF MINGW64 /c
$ git clone https://github.com/kyungskim-kit/PullRequestStudy.git
Cloning into 'PullRequestStudy'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), 38.01 KiB | 5.43 MiB/s, done.
Resolving deltas: 100% (2/2), done.
```

(예제) `git clone https://github.com/kyungskim-kit/PullRequestStudy.git`

② Clone

- “**git remote add**” 명령을 이용하여 방금 clone된 target repository를 원본 원격 저장소와도 연결해준다.

git remote add [새로운 원격저장소 별명]

[https://github.com/[원본계정 ID]/방금 복사한 원격 계정 주소]

```
KyungSooKim@DESKTOP-M68HRBF MINGW64 /c
$ cd PullRequestStudy

KyungSooKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (main)
$ git remote add real-PullRequestStudy https://github.com/kyungskim8476/PullRequestStudy.git
```


② Clone

git remote add [새로운 원격저장소 별명]

[https://github.com/[원본계정 ID]/방금 복사한 원격 계정 주소]

```
KyungSooKim@DESKTOP-M68HRBF MINGW64 /c
$ cd PullRequestStudy

KyungSooKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (main)
$ git remote add real-PullRequestStudy https://github.com/kyungskim8476/PullRequestStudy.git
```

- 이 작업을 수행하는 이유
 - 다른 사람의 원격 저장소를 fork한 결과, 이것이 내 GitHub 계정으로 복제되었음.
 - 이때, 나의 GitHub 계정에 복제된 원격 저장소를 clone하면, 내 로컬 컴퓨터에는 이와 연결된 workspace(local repository)가 생성됨.
 - 이 workspace는 나의 GitHub 계정에 복제된 원격 저장소하고만 연결되어 있음.
 - 따라서, 이를 본래 원격 저장소(즉, 타 사용자가 소유중인)와도 연결해 줄 필요가 있음.

② Clone

- 앞서 수행한 원격 저장소 연결 결과를 확인하기 위해 “**git remote -v**” 명령을 수행한다.

```
KyungSookKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (main)
$ git remote -v
origin https://github.com/kyungskim-kit/PullRequestStudy.git (fetch)
origin https://github.com/kyungskim-kit/PullRequestStudy.git (push)
real-PullRequestStudy https://github.com/kyungskim8476/PullRequestStudy.git (fetch)
real-PullRequestStudy https://github.com/kyungskim8476/PullRequestStudy.git (push)
```

- 설명
 - 원격 저장소 별명이 “**origin**”
 - 나의 계정에 업로드된 forked target remote repository.
 - 원격 저장소 별명이 “**real-PullRequestStudy**”
 - 관리자 계정의 original remote repository

③ 새로운 branch 생성

- 자신의 로컬 컴퓨터에서 독립적으로 작업을 수행할 수 있도록 새로운 branch를 생성한다.

```
git checkout -b [새로운 브랜치 이름]
```

```
KyungSooKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (main)  
$ git checkout -b newbranch  
Switched to a new branch 'newbranch'
```

(예제) `git checkout -b newbranch`

③ 새로운 branch 생성

- 새로운 브랜치 생성 후 현재 브랜치 리스트를 확인한다.

git branch

```
KyungSookKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (newbranch)
$ git branch
main
* newbranch
```

- “**git branch**” : 현재 브랜치의 리스트를 화면에 출력한다.
 - 현재 작업중인 브랜치는 앞에 * 가 표시된다.

④ Branch에서의 작업

- 이제 새로운 브랜치에서 코드를 자유롭게 수정하거나 삭제 또는 새로운 코드를 작성할 수 있다.

```
GNU nano 6.4      max_profit.py      Modified
# This code computes a maximum profit value...
import sys

def maxProfit_bruteforce (prices):
    max_price = -sys.maxint
    for i, price in enumerate(prices):
        for j in range(i, len(prices)):
            max_price = max(prices[j] - price, max_price)

    return max_price

^G Help      ^O Write Out ^W Where Is ^K Cut      ^T Execute
^X Exit      ^R Read File ^\ Replace  ^U Paste    ^J Justify
```

⑤ Add, commit, push 수행

- 코드 편집이 완료되면 **add**와 **commit** 명령(또는 **commit -a**)을 수행하여 자신의 로컬 저장소에 수정 사항을 반영한다.

```
KyungSooKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (newbranch)
$ git status
On branch newbranch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.c
        modified:   max_profit.py

no changes added to commit (use "git add" and/or "git commit -a")
```

⑤ Add, commit, push 수행

- 코드 편집이 완료되면 **add**와 **commit** 명령(또는 **commit -a**)을 수행하여 자신의 로컬 저장소에 수정 사항을 반영한다.

```
KyungSookKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (newbranch)
$ git add hello.c max_profit.py

KyungSookKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (newbranch)
$ git commit -m "modify hello.c and max_profit.py by kyungskim-kit"
[newbranch 8efbcb3] modify hello.c and max_profit.py by kyungskim-kit
2 files changed, 9 insertions(+), 6 deletions(-)

KyungSookKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (newbranch)
$ git status
On branch newbranch
nothing to commit, working tree clean
```

(예제) `git add hello.c max_profit.c`
`git commit -m "modify hello.c and max_profit.py"`

⑤ Add, commit, push 수행

- 이후, **push** 명령을 진행한다. 이때, 브랜치 이름을 반드시 지정해야 한다.

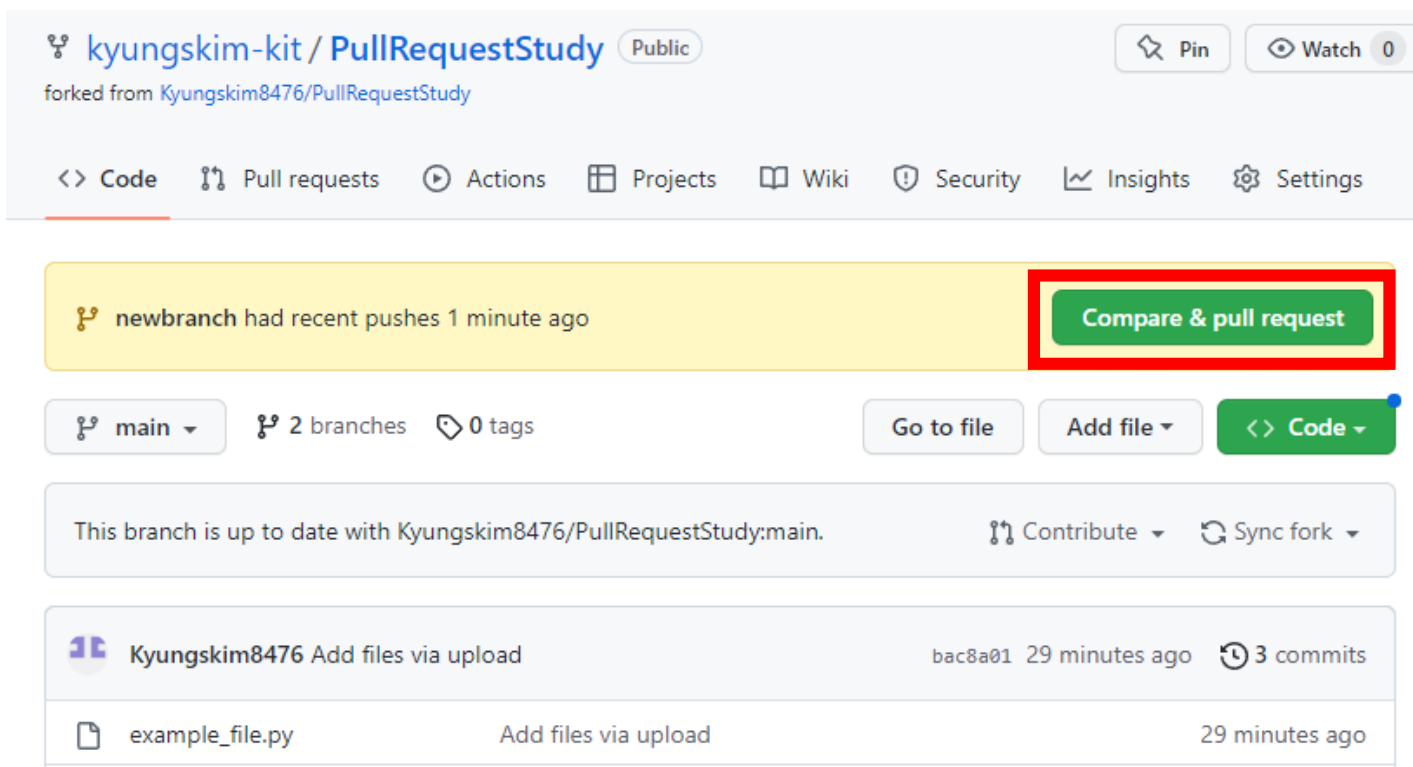
git push origin [브랜치 이름]

```
KyungSookKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (newbranch)
$ git push origin newbranch
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 20 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 684 bytes | 684.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'newbranch' on GitHub by visiting:
remote:   https://github.com/kyungskim-kit/PullRequestStudy/pull/new/newbranch
remote:
```

(예제) git push origin newbranch

⑥ Pull request 생성

- **push** 완료 후 본인 계정의 원격 저장소에 접속하면, “Compare & pull request” 버튼이 활성화되어 있음을 확인할 수 있다.
- 상기 버튼을 클릭하여 메시지를 작성하고 **pull request**를 생성한다.



⑥ Pull request 생성


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: Kyungskim8476/PullRequestSt...
base: main

head repository: kyungskim-kit/PullRequestStudy

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

modify hello.c and max_profit.py by kyungskim-kit

WritePreview

H B I  < >     @  

Modify the below files:
- hello.c
- max_profit.py.
Modifier: kyungskim-kit

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

☒ Allow edits by maintainers ?

Create pull request

⑥ Pull request 생성

Commits on Nov 21, 2022

modify hello.c and max_profit.py by kyungskim-kit

 kyungskim-kit committed 5 minutes ago



8efbcb3





Showing 2 changed files with 9 additions and 6 deletions.

Split

Unified

7  hello.c 

```
...    ...    @@ -1,13 +1,14 @@
1      1      #include <studio.h>
2      2      + #include <stdlib.h>
3      3
4      4      int main(void){
5      5          int i=0;
6      6          int sum=0;
7      7      -   printf("Hello World!!\n");
8      8      +   printf("Summation from 0 to 10\n");
9      9          for (i=0; i<10; i++){
10     10              sum += i;
11     11      -   return sum;
12     12      +   printf("The result value is %d\n", sum);
13     13      +   return 0;
14     14      }
```

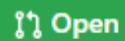
8  max_profit.py 

```
...    ...    @@ -1,9 +1,11 @@
1      1      - def maxProfit_bruteforce (prices):
2      2      -     max_price = 0
3      3
4      4      + # This code computes a maximum profit value...
5      5      + import sys
6      6
7      7      + def maxProfit_bruteforce (prices):
8      8      +     max_price = -sys.maximum
9      9
10     10      for i, price in enumerate(prices):
11     11          for j in range(i, len(prices)):
12     12              max_price = max(prices[j] - price, max_price)
13     13
14     14      -   return
15     15      +   return max_price
```

⑥ Pull request 생성

- “Conversation” 탭에서는 관리자와 기여자가 수정 사항에 대해 주고받은 의견(feedback)을 확인할 수 있다.

modify hello.c and max_profit.py by kyungskim-kit #1



kyungskim-kit wants to merge 1 commit into `Kyungskim8476:main` from `kyungskim-kit:newbranch`



Conversation 0



Commits 1



Checks 0



Files changed 2



kyungskim-kit commented 41 seconds ago



Modify the below files:

- hello.c
 - max_profit.py.
- Modifier: kyungskim-kit




modify hello.c and max_profit.py by kyungskim-kit


8efbcb3

⑥ Pull request 생성


- “Commits” 탭에서는 관리자와 기여자가 **commit**한 리스트를 확인할 수 있다.

modify hello.c and max_profit.py by kyungskim-kit #1


 Open


kyungskim-kit wants to merge 1 commit into [Kyungskim8476:main](#) from [kyungskim-kit:newbranch](#) 

 Conversation 0

 Commits 1

 Checks 0

 Files changed 2

 Commits on Nov 21, 2022

modify hello.c and max_profit.py by kyungskim-kit



kyungskim-kit committed 14 minutes ago

⑥ Pull request 생성

- “File changed” 탭에서는 현재 원격 저장소에서 어떠한 부분이 변경되었는지 세부적으로 확인할 수 있다.

modify hello.c and max_profit.py by kyungskim-kit #1

[Open](#) kyungskim-kit wants to merge 1 commit into [Kyungskim8476:main](#) from [kyungskim-kit:newbranch](#)

Conversation 0

Commits 1

Checks 0

Files changed 2

Changes from all commits File filter Conversations

Filter changed files

hello.c

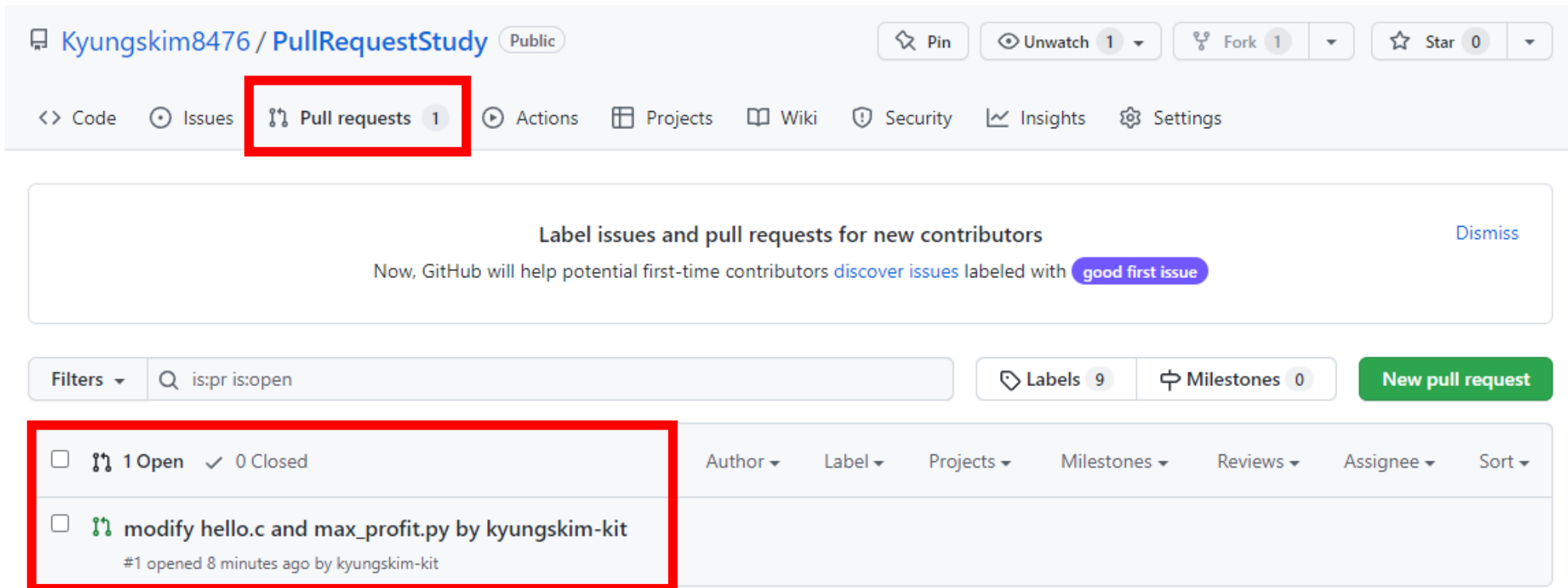
max_profit.py

7 hello.c

```
... @@ -1,13 +1,14 @@
1 1 #include <studio.h>
2 2 + #include <stdlib.h>
3 3
4 4 int main(void){
5 5     int i=0;
6 6     int sum=0;
7 7 -
7 7 - printf("Hello World!!\n");
```

⑦ Review & merge

- Pull request를 받은 target repository의 관리자는 코드의 수정 사항들을 확인한 후 이를 반영할 것인가(즉, **merge** 여부)를 결정한다.




The screenshot shows the GitHub interface for the repository 'Kyungskim8476 / PullRequestStudy'. The 'Pull requests' tab is selected and highlighted with a red box. Below the repository header, there is a notification bar about labeling issues and pull requests for new contributors. The search bar shows 'is:pr is:open'. The pull request list shows one open pull request, 'modify hello.c and max_profit.py by kyungskim-kit', which is also highlighted with a red box. The pull request was opened 8 minutes ago by kyungskim-kit.

modify hello.c and max_profit.py by kyungskim-kit #1

[Open](#) kyungskim-kit wants to merge 1 commit into [kyungskim8476:main](#) from [kyungskim-kit:newbranch](#)

Conversation 0 Commits 1 Checks 0 Files changed 2



kyungskim-kit commented 9 minutes ago
 First-time contributor

Modify the below files:

- hello.c
- max_profit.py.


Modifier: kyungskim-kit

modify hello.c and max_profit.py by kyungskim-kit

Looks like you got your first contribution! Visit your community profile to learn more about recommended open source practices.

[Go to community profile](#)


Add more commits by pushing to the **newbranch** branch on [kyungskim-kit/PullRequestStudy](#).



Require approval from specific reviewers before merging


[Branch protection rules](#) ensure specific people approve pull requests before they're merged.

[Add rule](#)



Continuous integration has not been set up

[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.



This branch has no conflicts with the base branch

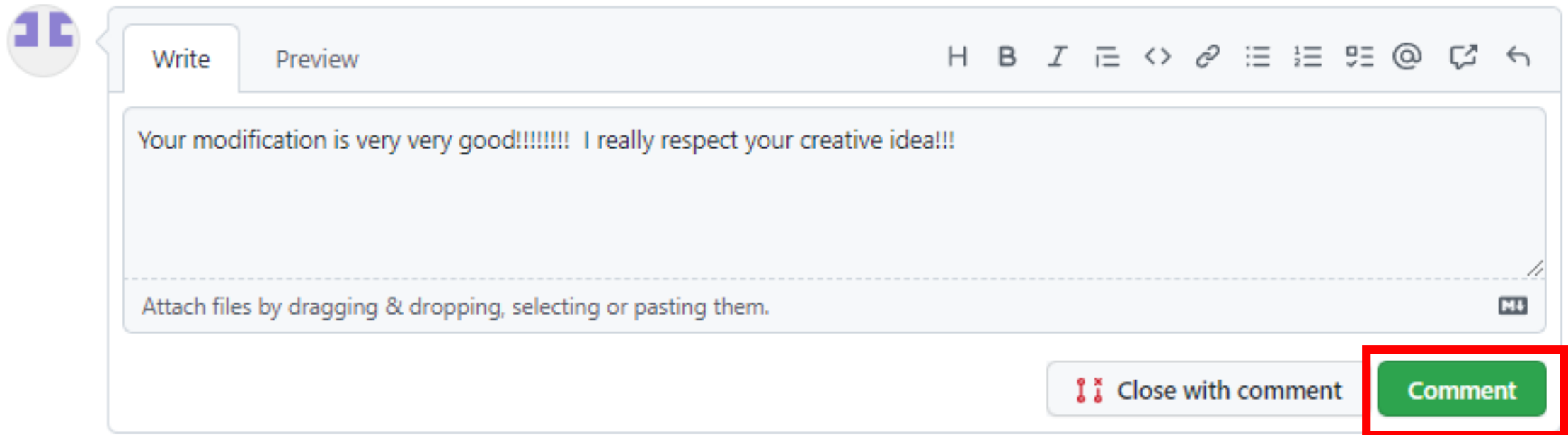
Merging can be performed automatically.

Merge pull request

You can also [open this in GitHub Desktop](#) or [view command line instructions](#).

⑦ Review & merge


- “Write” 탭에서 기여자가 수정한 내용에 대해 코멘트를 작성한 후 “Comment” 버튼을 클릭한다.




Write Preview

H B I ≡ <> 🔗 ≡ ≡ ≡ @ ↻ ↶

Your modification is very very good!!!!!! I really respect your creative idea!!!

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

 Close with comment **Comment**

⑦ Review & merge

- 기여자가 수정한 내용을 승인하는 경우 “Confirm merge” 버튼을 클릭하여 기여자가 수정한 내용을 현재 브랜치에 합병한다.

Add more commits by pushing to the `newbranch` branch on `kyungskim-kit/PullRequestStudy`.



Merge pull request #1 from `kyungskim-kit/newbranch`

modify `hello.c` and `max_profit.py` by `kyungskim-kit`

This commit will be authored by `97086141+Kyungskim8476@users.noreply.github.com`

Confirm merge

Cancel


⑦ Review & merge


- 그렇다면, “Conversation” 탭에 관리자가 작성한 feedback 내용이 나타난다.

modify hello.c and max_profit.py by kyungskim-kit #1

Merged Kyungskim8476 merged 1 commit into [Kyungskim8476:main](#) from [kyungskim-kit:newbranch](#) 32 seconds ago

Conversation 0 Commits 1 Checks 0 Files changed 2






kyungskim-kit commented 12 minutes ago Contributor  ...


Modify the below files:

- hello.c
- max_profit.py.

Modifier: kyungskim-kit

  modify hello.c and max_profit.py by kyungskim-kit 8efbcb3

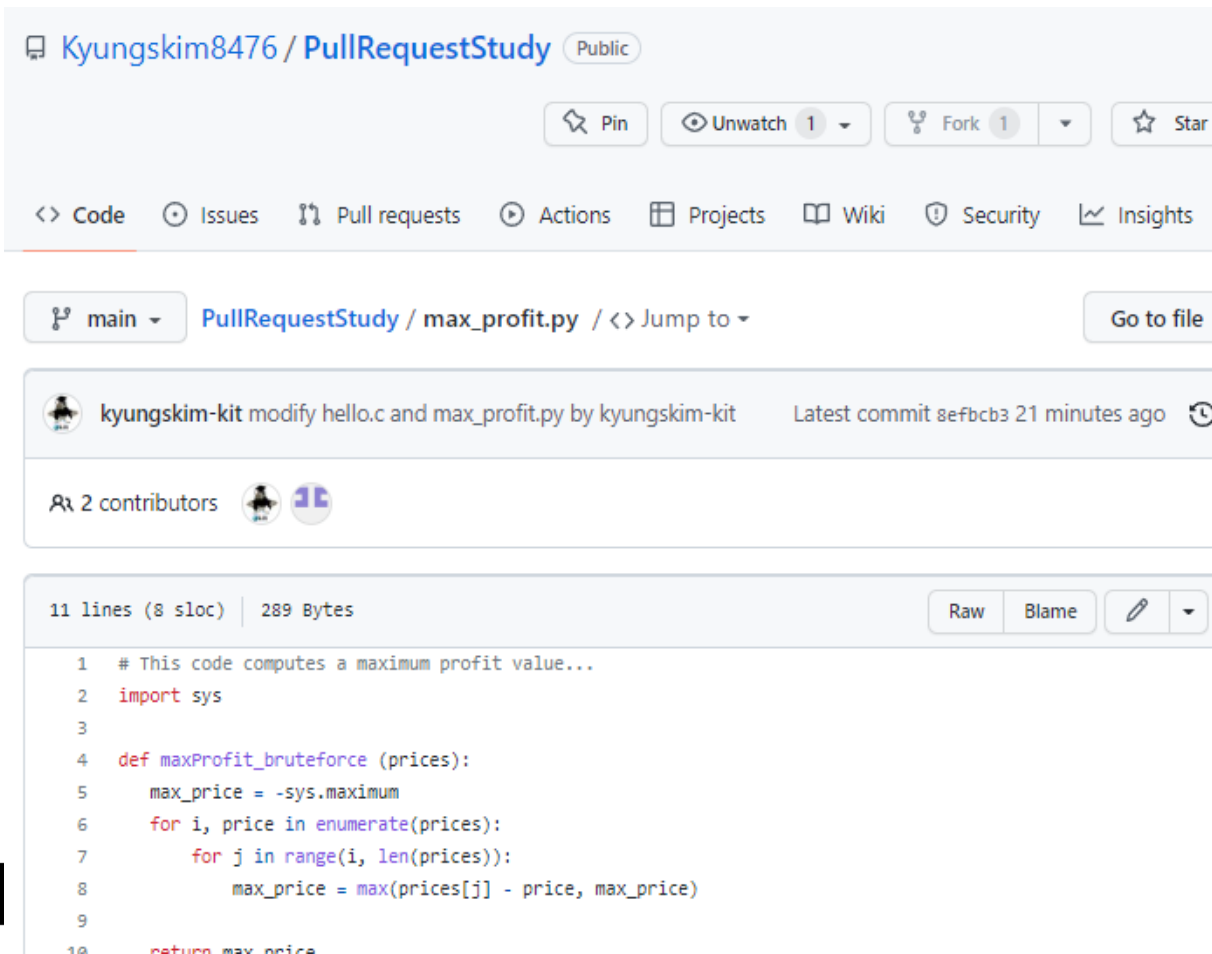


Kyungskim8476 commented 1 minute ago Owner  ...

Your modification is very very good!!!!!! I really respect your creative idea!!!

⑦ Review & merge

- 이후 관리자의 원격 저장소를 확인해보면 기여자가 수정한 내용이 정상적으로 반영되었음을 확인할 수 있다.



Kyungskim8476 / PullRequestStudy Public

Pin Unwatch 1 Fork 1 Star

Code Issues Pull requests Actions Projects Wiki Security Insights

main PullRequestStudy / max_profit.py / <> Jump to Go to file

kyungskim-kit modify hello.c and max_profit.py by kyungskim-kit Latest commit 8efbcb3 21 minutes ago

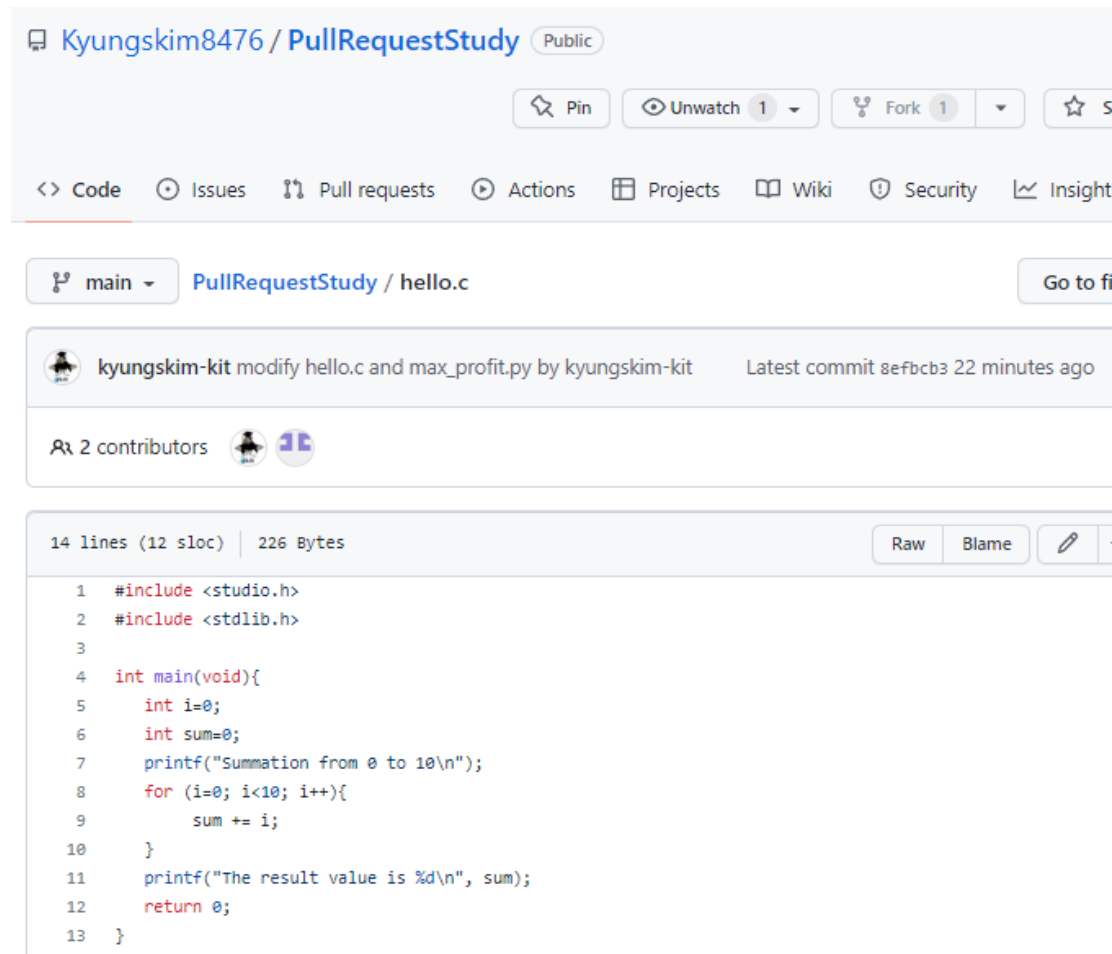
2 contributors

11 lines (8 sloc) 289 Bytes

```

1 # This code computes a maximum profit value...
2 import sys
3
4 def maxProfit_bruteforce (prices):
5     max_price = -sys.maximum
6     for i, price in enumerate(prices):
7         for j in range(i, len(prices)):
8             max_price = max(prices[j] - price, max_price)
9
10     return max_price

```



Kyungskim8476 / PullRequestStudy Public

Pin Unwatch 1 Fork 1 Star

Code Issues Pull requests Actions Projects Wiki Security Insights

main PullRequestStudy / hello.c Go to file

kyungskim-kit modify hello.c and max_profit.py by kyungskim-kit Latest commit 8efbcb3 22 minutes ago

2 contributors

14 lines (12 sloc) 226 Bytes

```

1 #include <studio.h>
2 #include <stdlib.h>
3
4 int main(void){
5     int i=0;
6     int sum=0;
7     printf("Summation from 0 to 10\n");
8     for (i=0; i<10; i++){
9         sum += i;
10    }
11    printf("The result value is %d\n", sum);
12    return 0;
13 }

```

추가적인 기여를 수행하는 경우

- Target repository에 대하여 추가적으로 기여하고자 하는 경우, “**pull**” 명령을 사용하여 target repository와 현재 브랜치를 동기화한 후 앞서 설명한 작업들을 수행한다.

➤ 동기화 명령:

```
git pull [원격 저장소명]
```

target repository의 별명



추가적인 기여를 수행하는 경우

- Target repository에 대하여 추가적으로 기여하고자 하는 경우, “pull” 명령을 사용하여 target repository와 현재 브랜치를 동기화한 후 앞서 설명한 작업들을 수행한다.

(예제) `git pull real-PullRequestStudy`

```
KyungSooKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (newbranch)
$ git pull real-PullRequestStudy
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 650 bytes | 650.00 KiB/s, done.
From https://github.com/kyungskim8476/PullRequestStudy
* [new branch]      main      -> real-PullRequestStudy/main
You asked to pull from the remote 'real-PullRequestStudy', but did not specify
a branch. Because this is not the default configured remote
for your current branch, you must specify a branch on the command line.
```

더 이상 기여를 수행하지 않는 경우

- 작업하던 로컬 저장소의 브랜치를 main으로 변경한 후 작업한 삭제한다.
 - 브랜치 변경: `git checkout [이동할 브랜치 이름]`
 - 브랜치 삭제: `git branch -D [삭제할 브랜치 이름]`

```
KyungSooKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (newbranch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

```
KyungSooKim@DESKTOP-M68HRBF MINGW64 /c/PullRequestStudy (main)
$ git branch -D newbranch
Deleted branch newbranch (was 8efbcb3).
```

(예제) `git checkout main`
`git branch -D newbranch`

Pull request 실습

Assignment #3 – Pull request 실습

- 본 강의에서 설명한 일련의 전 과정을 실습한 후 실습 결과를 보고서로 상세히 작성하여 LMS에 제출한다.
- 실습 시 조원 1명이 “관리자” 역할을 수행하면 나머지는 자동으로 “기여자”로서 작업을 수행함.
- 자신이 “관리자”로서 수행한 작업들과 “기여자”로서 작업한 화면들을 캡처하여 보고서에 반드시 첨부해야 한다.
 - 단, 자신이 “기여자”로서 실습한 결과는 “기여자 1”과 “기여자 2”의 작업에 대해서만 첨부할 것.

Assignment #3 – Pull request 실습

- 실습 시 유의사항

- ① 실습을 위한 GitHub 계정은 본 수업용 계정을 사용할 것.

- ② Commit 수행 시 commit 메시지에는 자신의 이름을 영어로 작성할 것.

- (예) `git commit -m “committed by kyung soo kim”`

- ③ 실습에 사용되는 코드는 프로그래밍 시간에 작성한 코드들을 사용할 것.

Q & A