제품소프트웨어 패키징
(Git과 GitHub를 활용한 소스코드 관리)

GitHub에서 Code Review 하기

한국기술교육대학교
온라인평생교육원

학습내용

- GitHub에서 Code Review 하기
- GitHub에서 Code Review 실습

학습목표

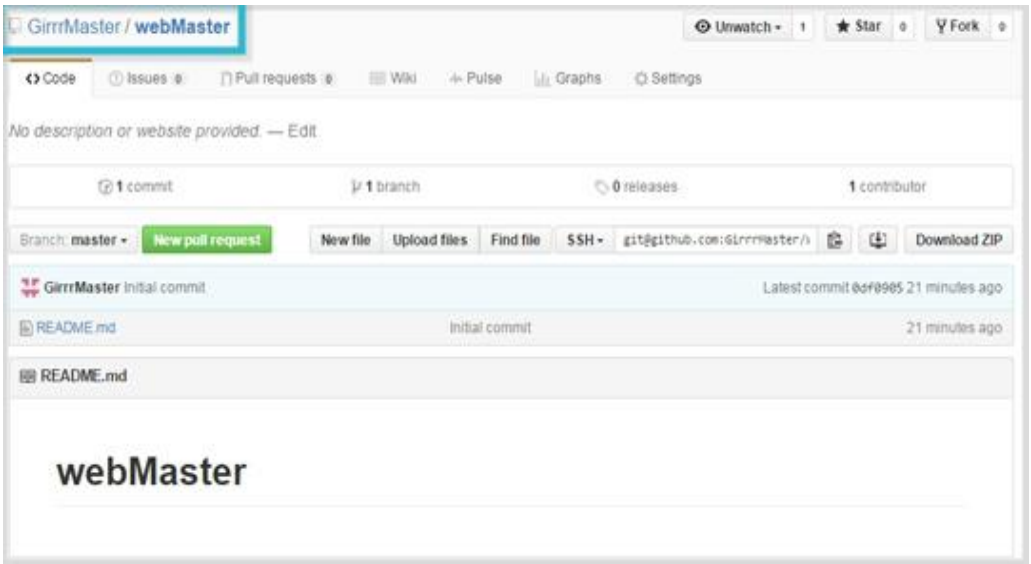
- GitHub에서 Code Review를 하기 위하여 설정을 할 수 있다.
- GitHub에서 소스코드를 Review하고 반영하는 Code Review 작업을 수행할 수 있다.

GitHub에서 Code Review 하기

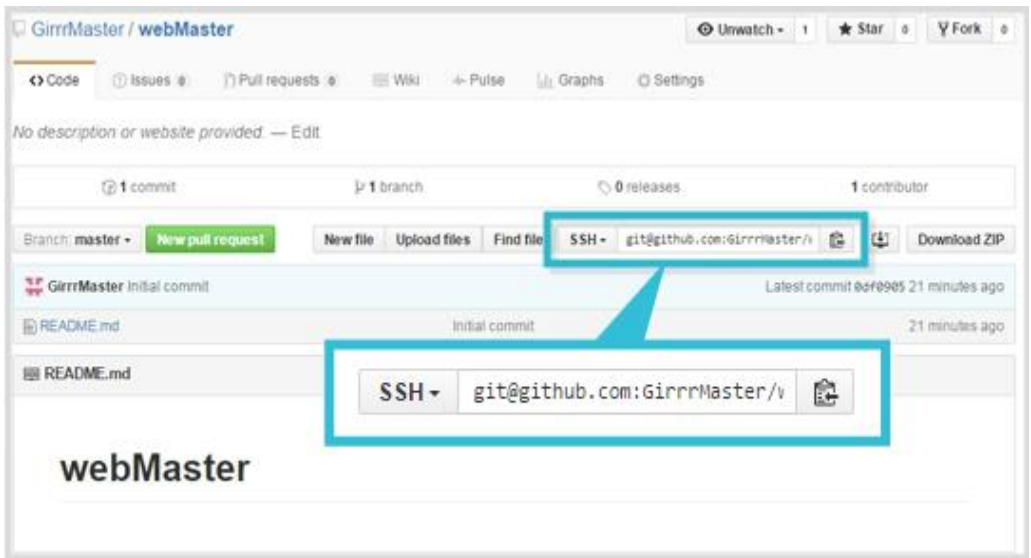
1. Code Review 준비

1) 로컬저장소 및 파일 생성

- webMaster 이름으로 원격저장소 생성



- 저장소의 ssh 방식의 주소 복사



- git을 이용하여 로컬 저장소에서 셋팅하고 GitHub에 반영

GitHub에서 Code Review 하기

1. Code Review 준비

1) 로컬저장소 및 파일 생성

- ssh 주소를 이용하여 git bash 창에서 clone

```
$ git clone git@github.com:GirrrMaster/webMaster.git
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
Checking connectivity... done.
```

- webMaster 저장소에서 가져온 저장소 상태 확인

```
user@userpc MINGW64 /c/gitFolder
$ cd webMaster/

user@userpc MINGW64 /c/gitFolder/webMaster (master)
$ ls
README.md
```

- 기본적인 폴더 구성 및 파일 생성

```
$ mkdir tag

$ touch index.html style.css

$ ls
index.html  README.md  style.css  tag/

$ cd tag

user@userpc MINGW64 /c/gitFolder/webMaster/tag (master)
$ touch form.html header.html table.html

form.html  header.html  table.html

$ cd ..

$ ls
index.html  README.md  style.css  tag/
```

GitHub에서 Code Review 하기

1. Code Review 준비

1) 로컬저장소 및 파일 생성

- 초기 셋팅을 저장소에 등록 후 확인

```
user@userpc MINGW64 /c/gitFolder/webMaster (master)
$ git add *

user@userpc MINGW64 /c/gitFolder/webMaster (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   index.html
        new file:   style.css
        new file:   tag/form.html
        new file:   tag/header.html
        new file:   tag/table.html
```

- commit 메시지를 입력 후 원격저장소에 등록

```
user@userpc MINGW64 /c/gitFolder/webMaster (master)
$ git commit -m "Setup Working Directory(girrrMaster)"
[master 1aaf2df] Setup Working Directory(girrrMaster)
 5 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
 create mode 100644 style.css
 create mode 100644 tag/form.html
 create mode 100644 tag/header.html
 create mode 100644 tag/table.html

user@userpc MINGW64 /c/gitFolder/webMaster (master)
$ git log
commit 1aaf2dfbbec7429cf035bf2a6c7256b40faa4571
Author: GirrrMaster <girrr.master@gmail.com>
Date:   Wed Mar 16 17:16:10 2016 +0900

    Setup Working Directory(girrrMaster)

commit 0df0905f7b5a24d3a2b3d28bf3113b12a501da06
Author: GirrrMaster <girrr.master@gmail.com>
Date:   Wed Mar 16 16:40:48 2016 +0900

    Initial commit
```

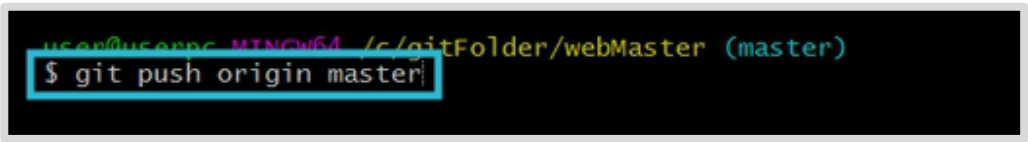
- commit 결과 확인

GitHub에서 Code Review 하기

1. Code Review 준비

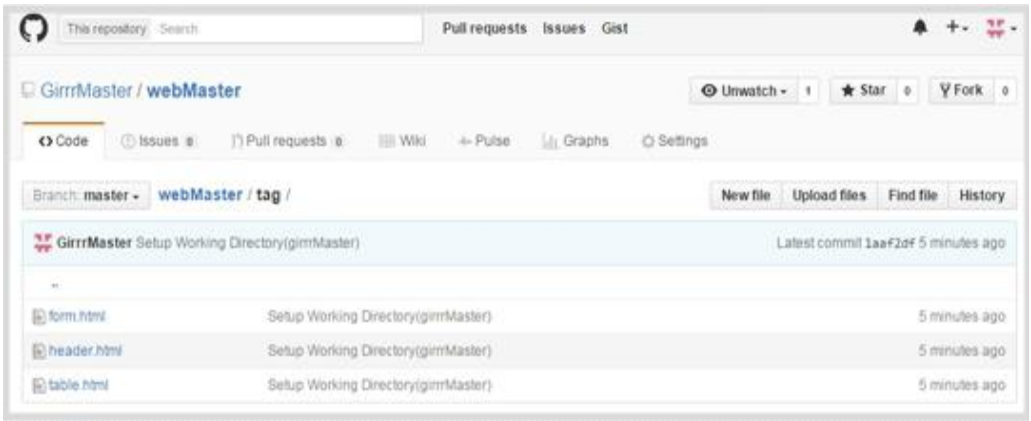
1) 로컬저장소 및 파일 생성

- commit 메시지를 입력 후 원격저장소에 등록



- webMaster 저장소의 master branch에 등록

- 원격저장소로 등록이 되었는지 GitHub에서 확인



1. Code Review 준비

2) 로컬저장소에서 branch 생성

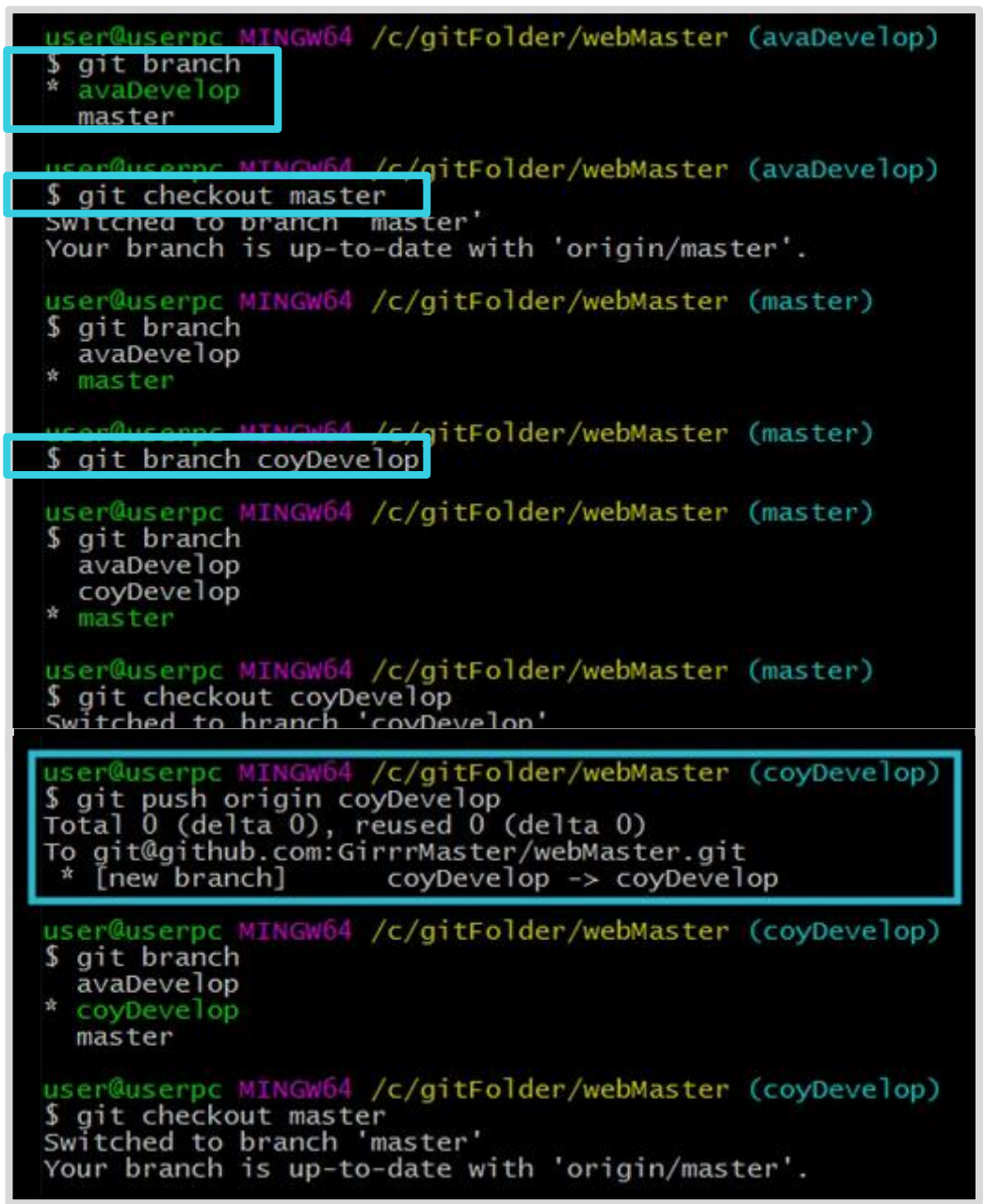
- branch 생성 - avaDevelop

GitHub에서 Code Review 하기

1. Code Review 준비

2) 로컬저장소에서 branch 생성

- branch 생성 - coyDevelop

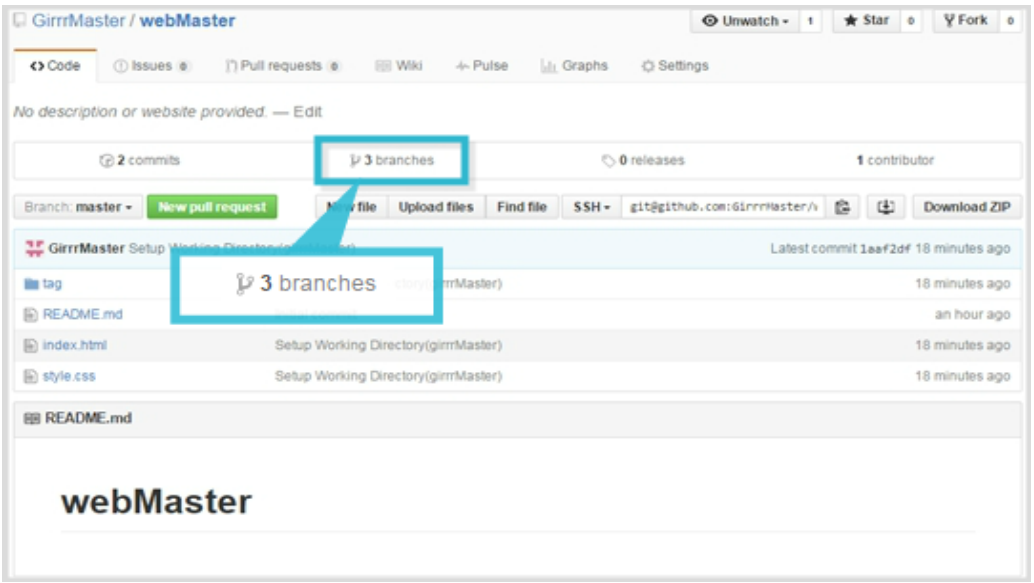


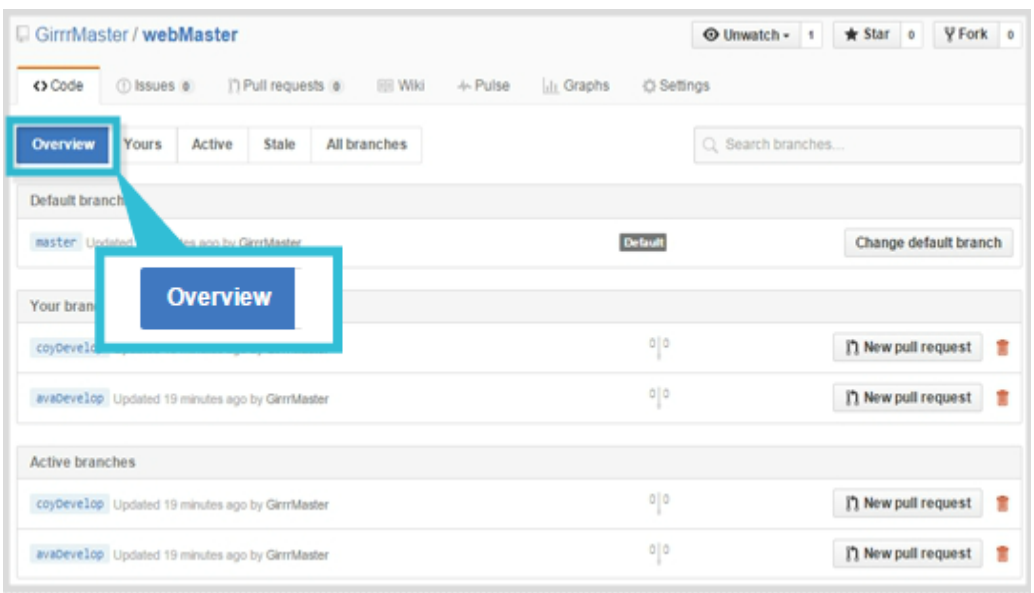
GitHub에서 Code Review 하기

1. Code Review 준비

2) 로컬저장소에서 branch 생성

- 원격저장소에 반영이 되었는지 확인

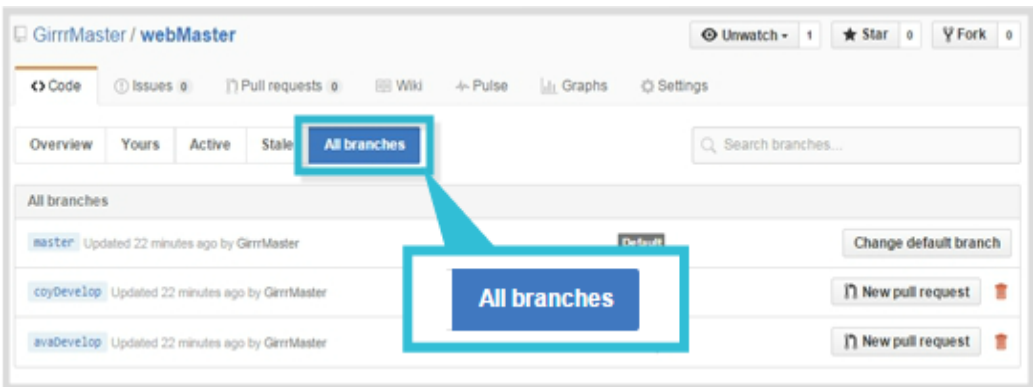




1. Code Review 준비

2) 로컬저장소에서 branch 생성

- 원격저장소에 반영이 되었는지 확인

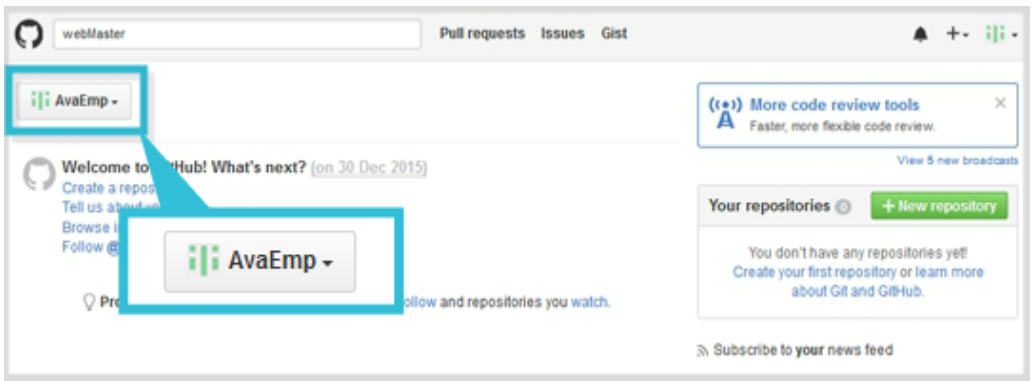


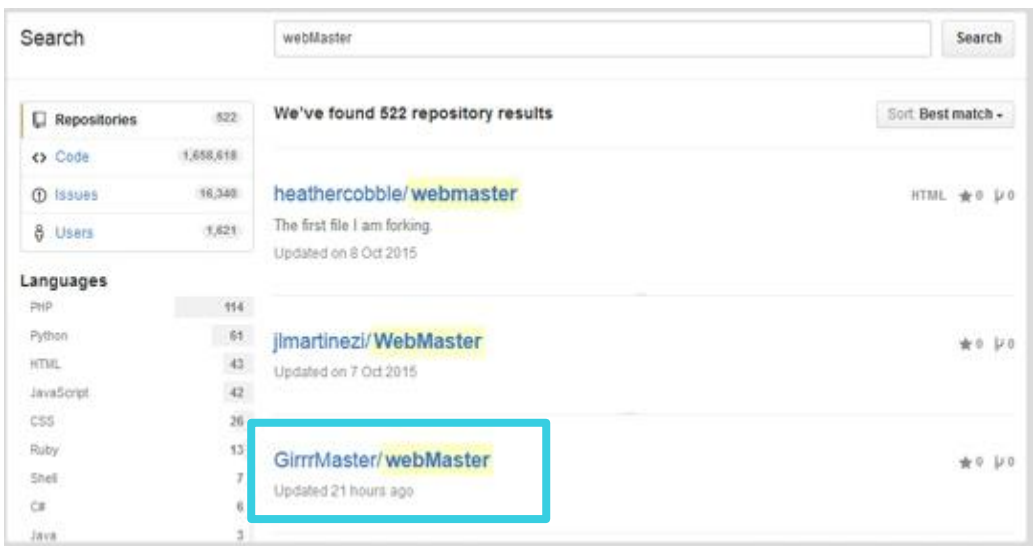
GitHub에서 Code Review 하기

1. Code Review 준비

3) Fork 기능 수행 및 Merge 완료

- 개발자1 계정에서 webMaster 저장소로 이동



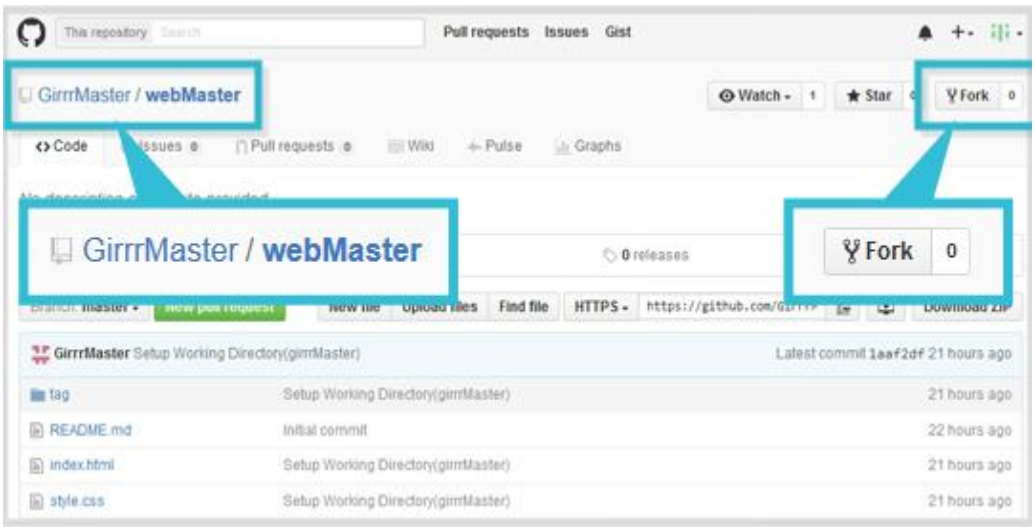


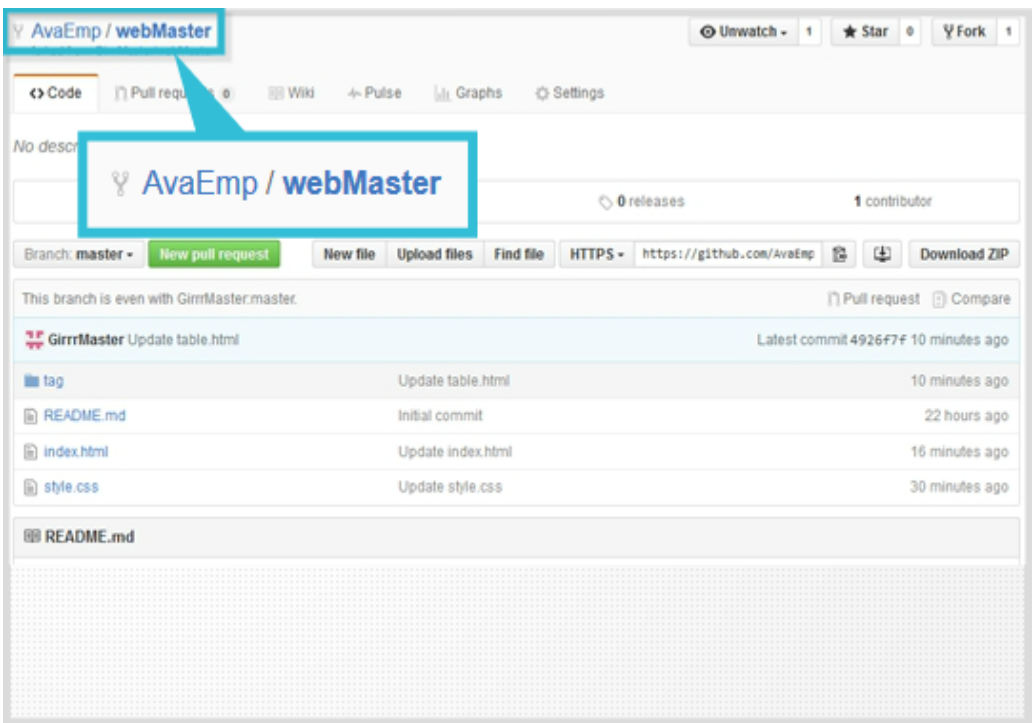
GitHub에서 Code Review 하기

1. Code Review 준비

3) Fork 기능 수행 및 Merge 완료

- girrrMaster의 webMaster 저장소를 Fork 하여 자신의 계정으로 가져옴



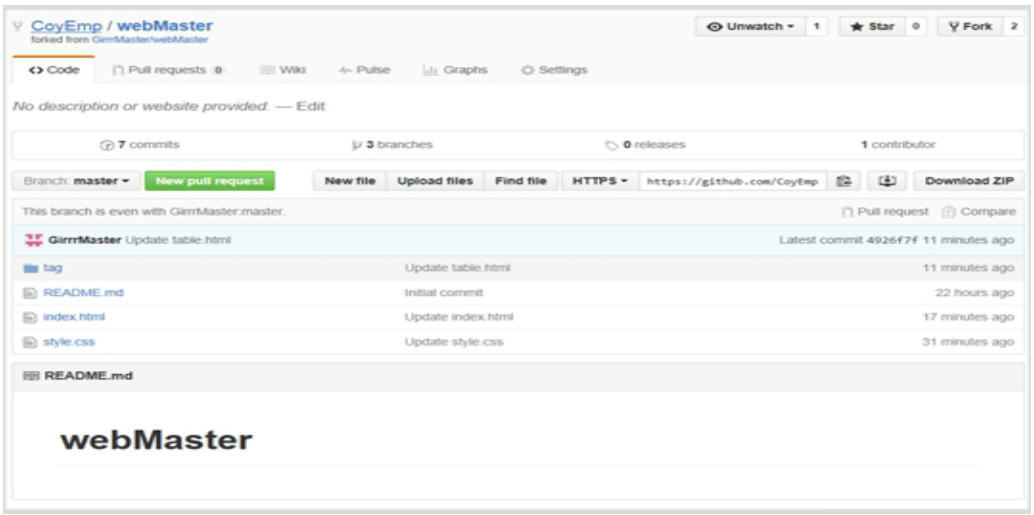


GitHub에서 Code Review 하기

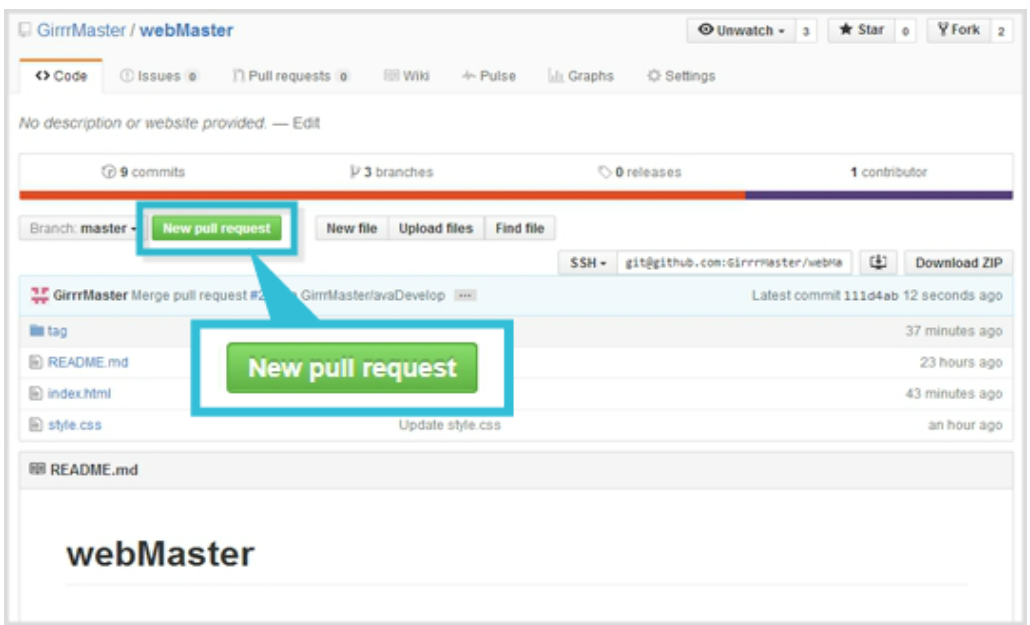
1. Code Review 준비

3) Fork 기능 수행 및 Merge 완료

- 개발자2 계정에서 webMaster 저장소를 fork



- New pull request 버튼 클릭
  - girrrMaster계정에서 avaDevelop, coyDevelop branch 로 master branch 의 소스코드를 복사 이동

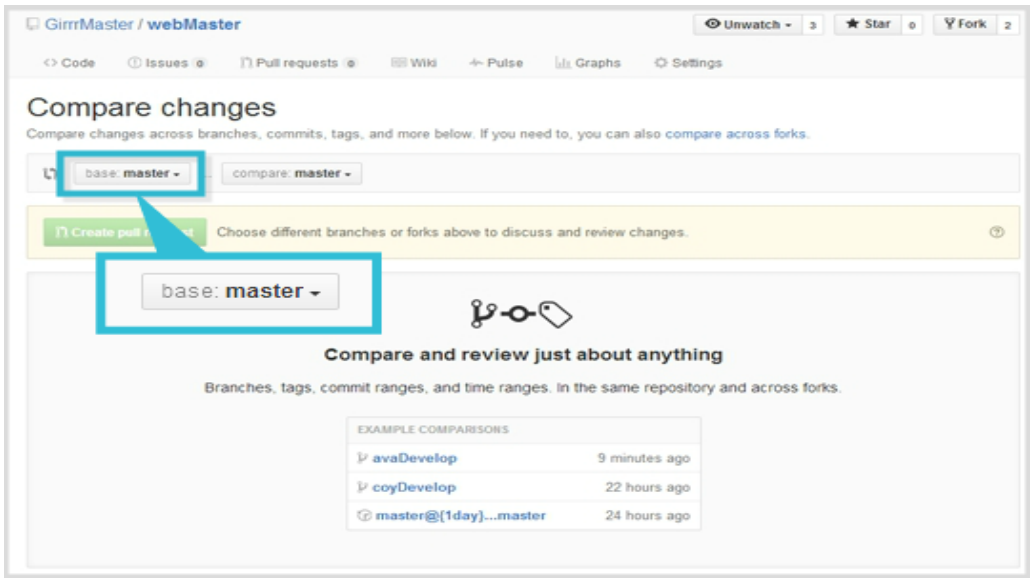


GitHub에서 Code Review 하기

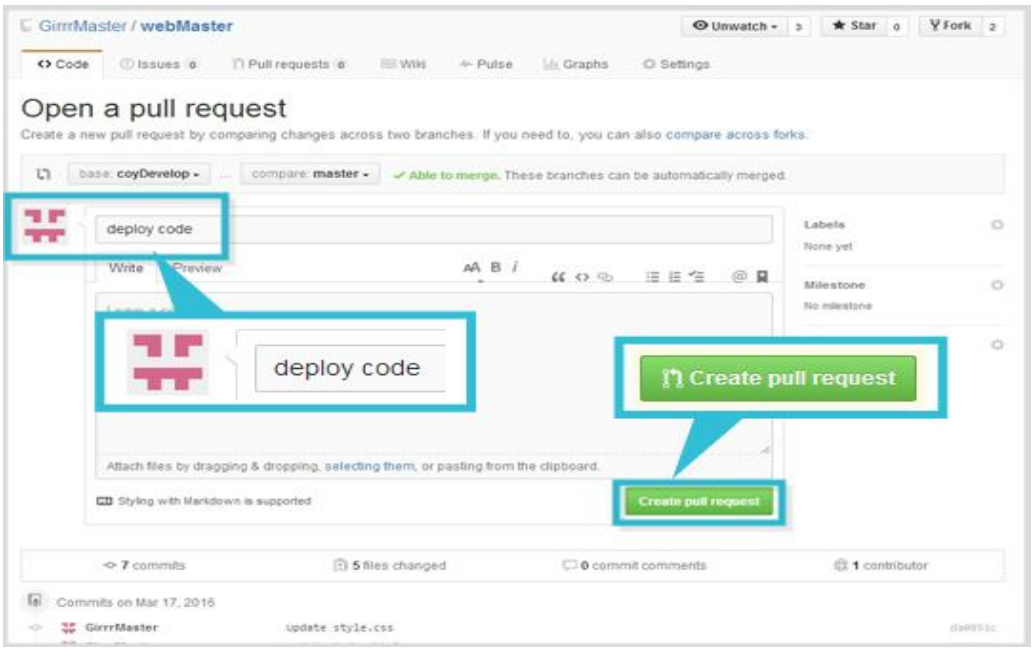
1. Code Review 준비

3) Fork 기능 수행 및 Merge 완료

- base 항목을 avaDevelop, coyDevelop branch로 바꿔가면서 pull request 실행



- 두 branch 간 비교된 내용을 확인하고 Create pull request 실행

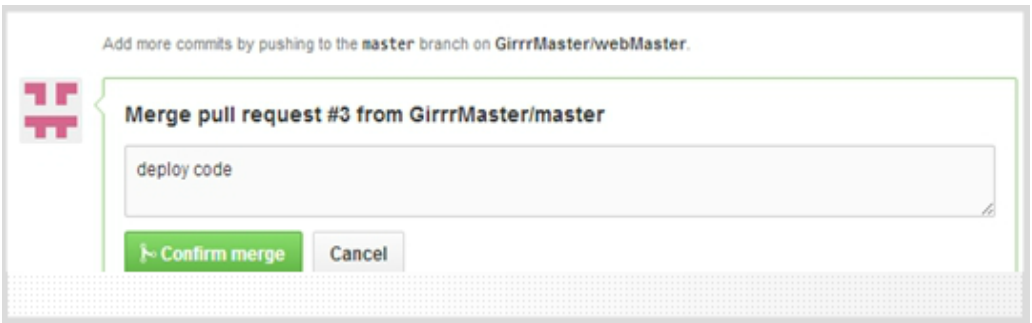


GitHub에서 Code Review 하기

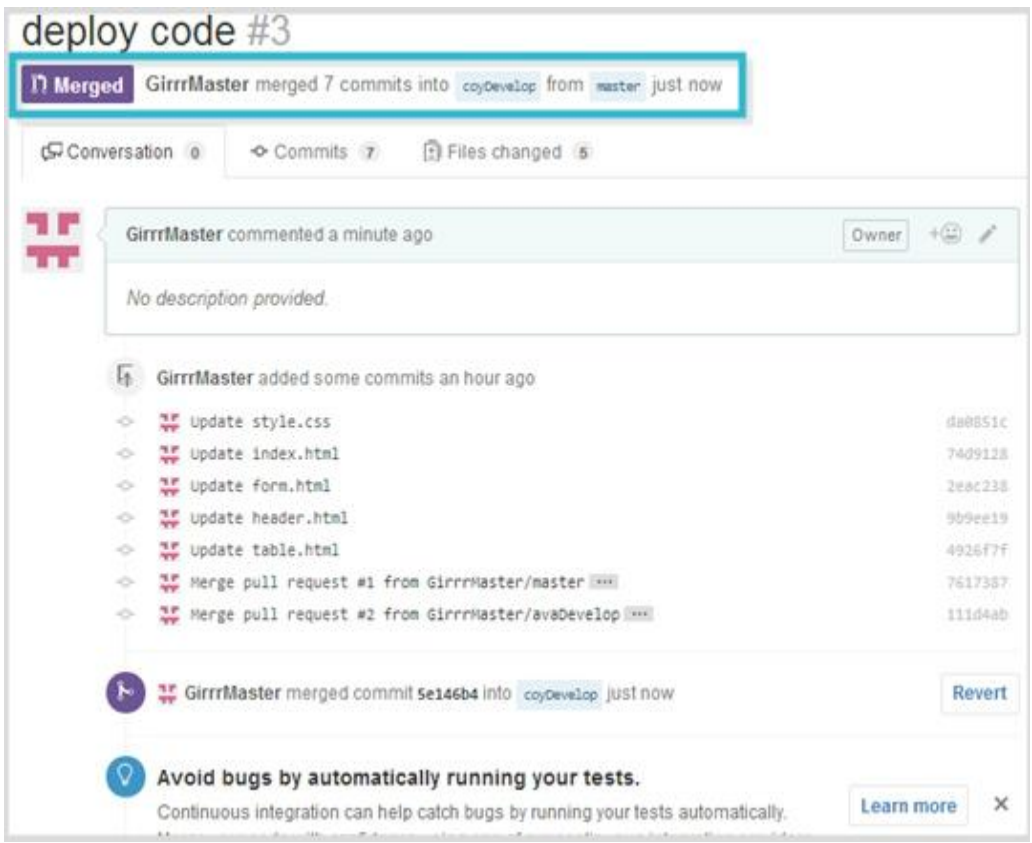
1. Code Review 준비

3) Fork 기능 수행 및 Merge 완료

- Confirm merge 완료된 창 확인



- Merge 완료된 창 확인



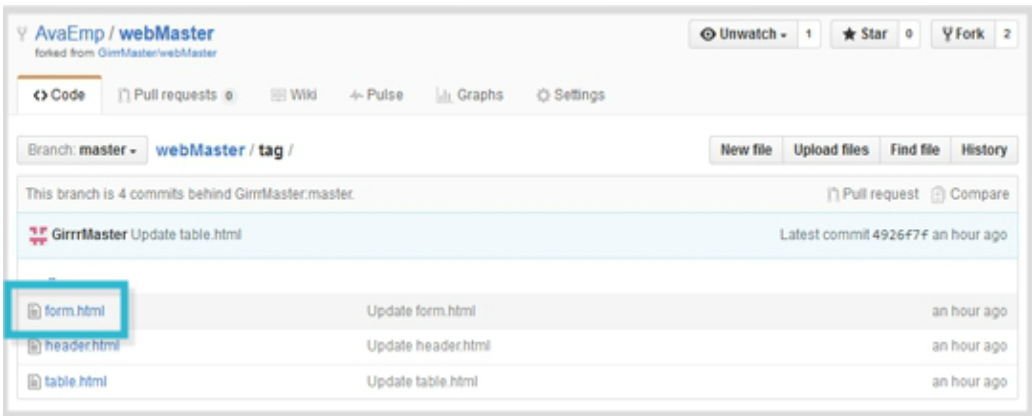
- 모두 동일한 파일

GitHub에서 Code Review 하기

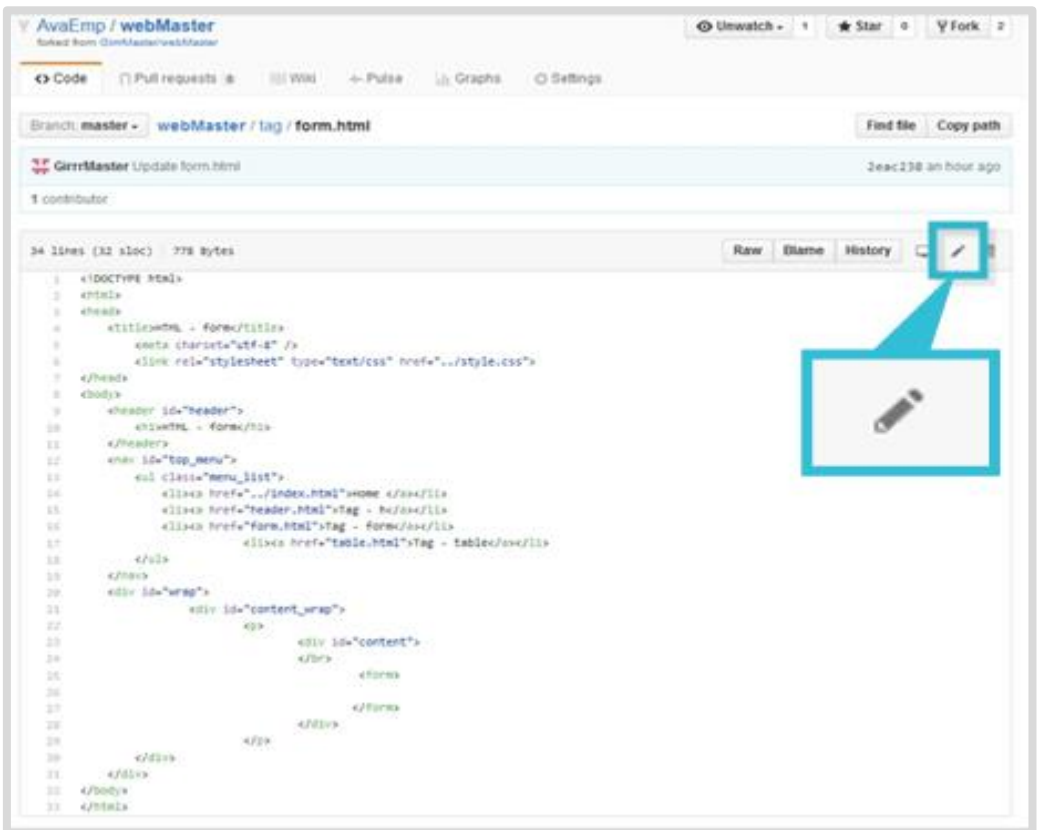
1. Code Review 준비

4) AvaEmp 계정에서 소스코드 업데이트

- avaEmp 개발자 계정에서 form.html 항목 클릭



- 화면 우측 부분에 수정 버튼 클릭

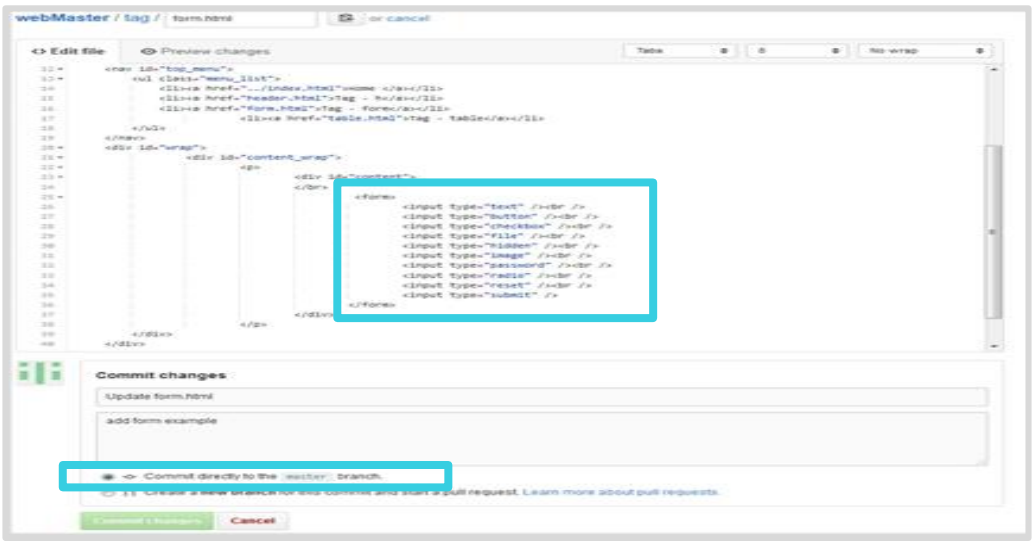


GitHub에서 Code Review 하기

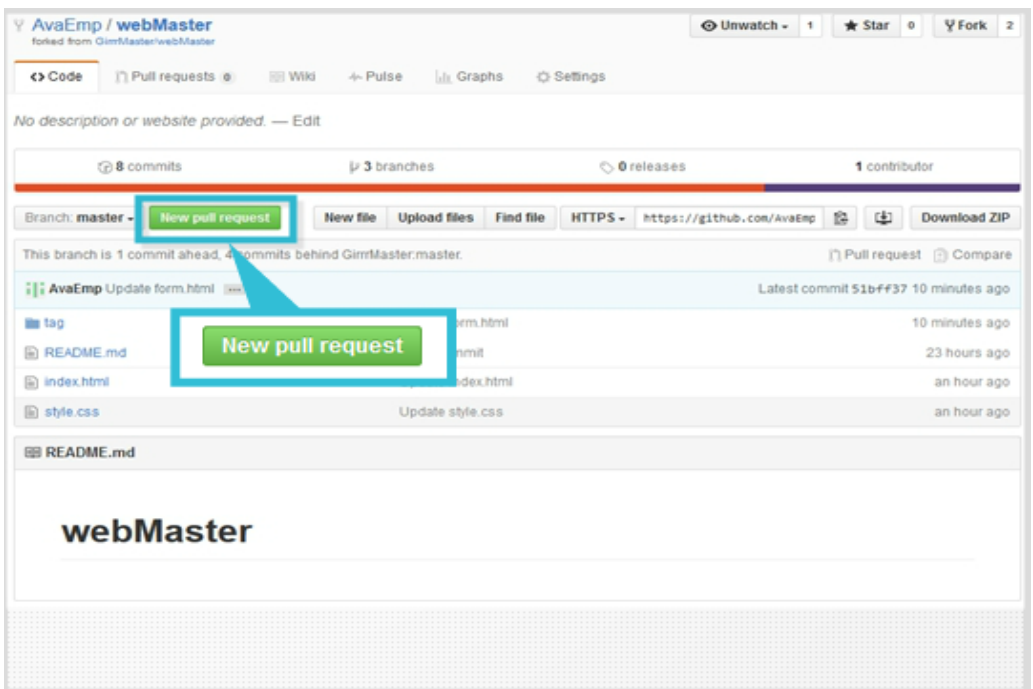
1. Code Review 준비

4) AvaEmp 계정에서 소스코드 업데이트

- html form 예제를 생성 추가 후 Commit



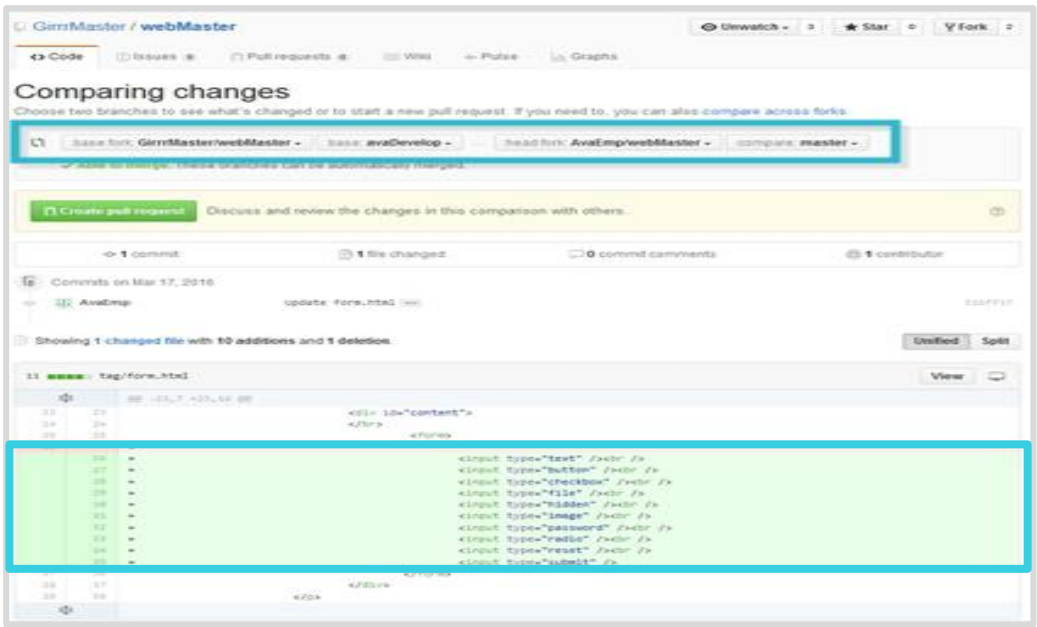
- pull request 요청



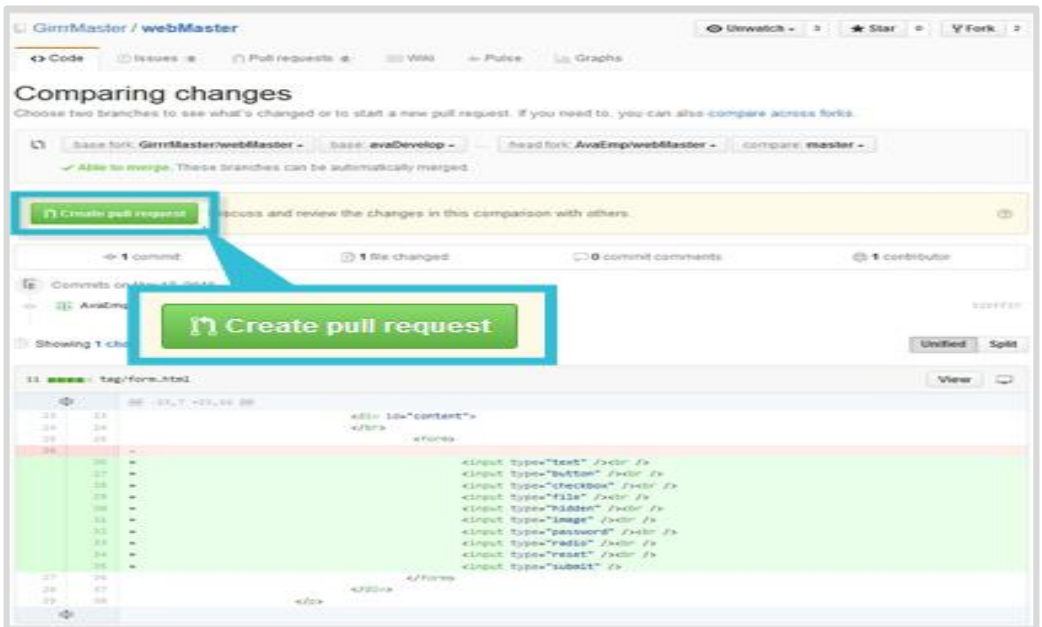
1. Code Review 준비

4) AvaEmp 계정에서 소스코드 업데이트

- 저장소를 비교하여 업데이트 된 내용이 보임



- Create pull request 버튼을 눌러 pull request 생성

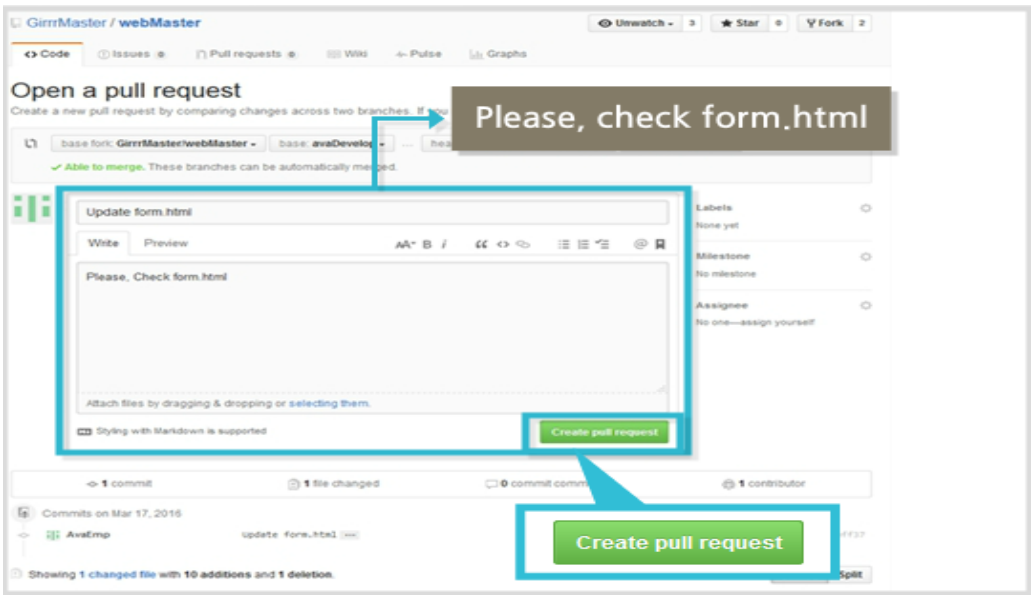


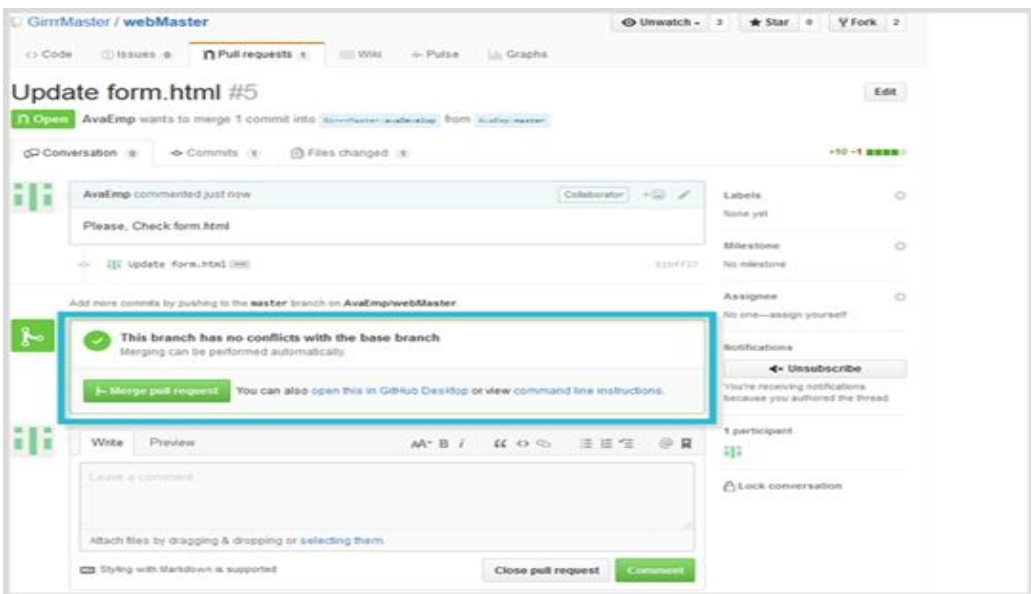
GitHub에서 Code Review 하기

1. Code Review 준비

4) AvaEmp 계정에서 소스코드 업데이트

- 요청 메시지 입력 후, Create pull request 버튼 클릭



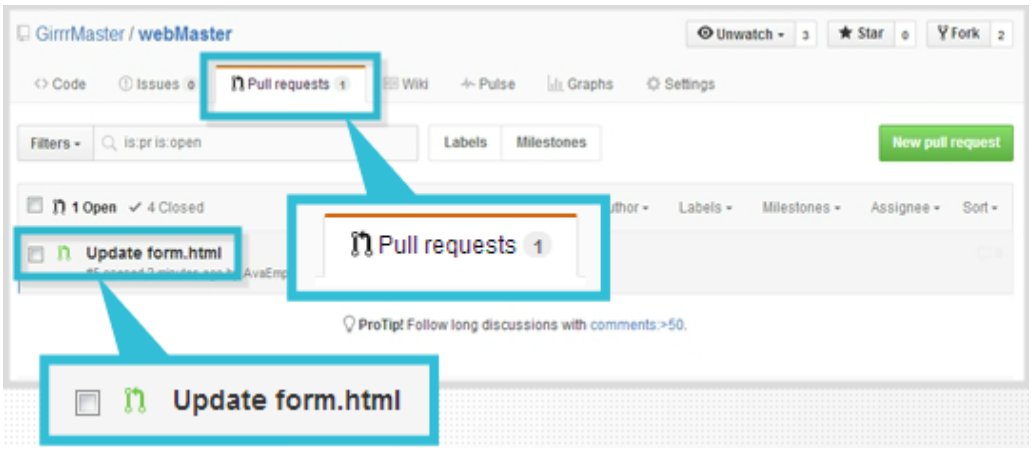


GitHub에서 Code Review 하기

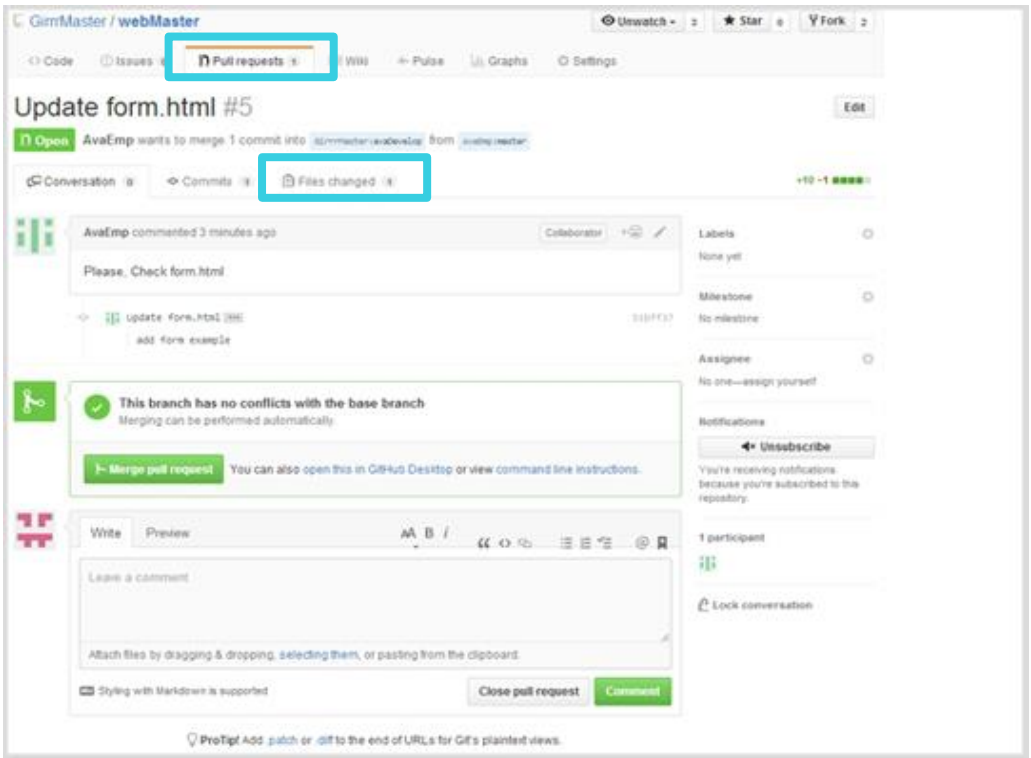
1. Code Review 준비

4) AvaEmp 계정에서 소스코드 업데이트

- Update form.html pull request 확인



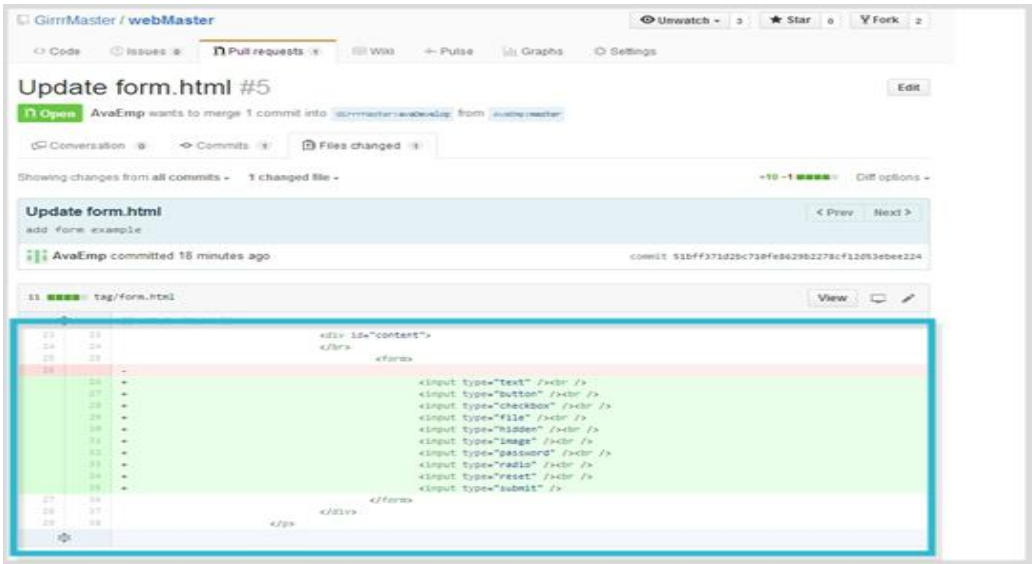
- files changed 탭을 눌러 수정된 부분 확인



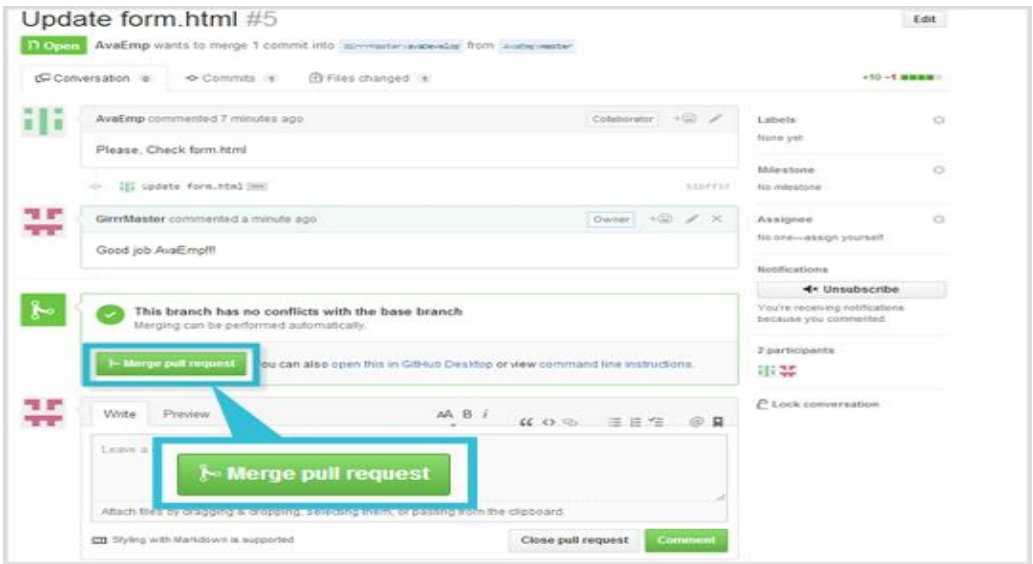
1. Code Review 준비

4) AvaEmp 계정에서 소스코드 업데이트

- files changed 탭을 눌러 수정된 부분 확인



- 확인했다는 메시지 입력 후 Merge pull request 수행

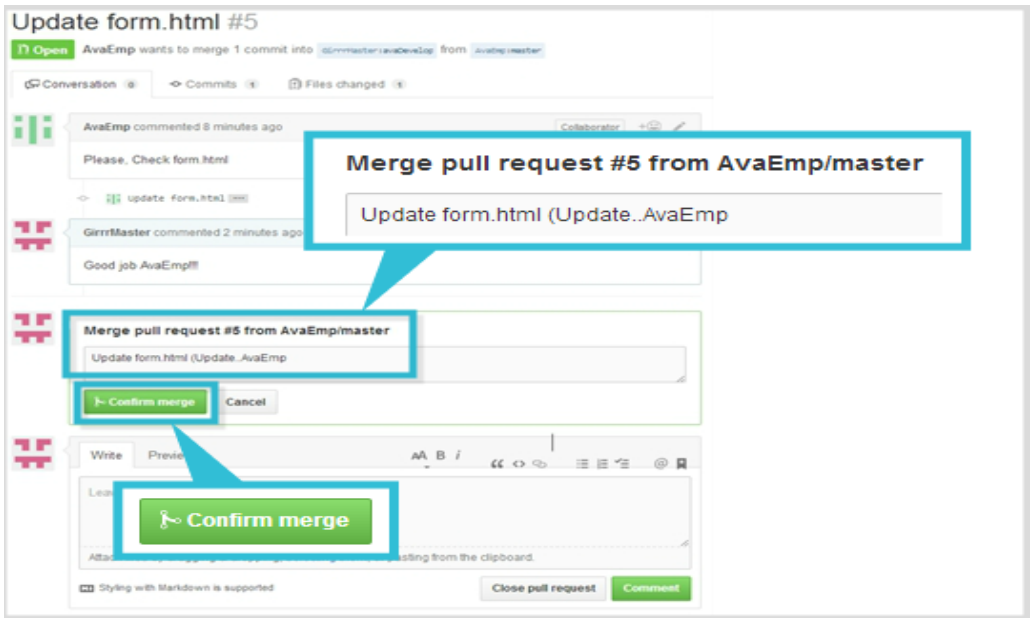


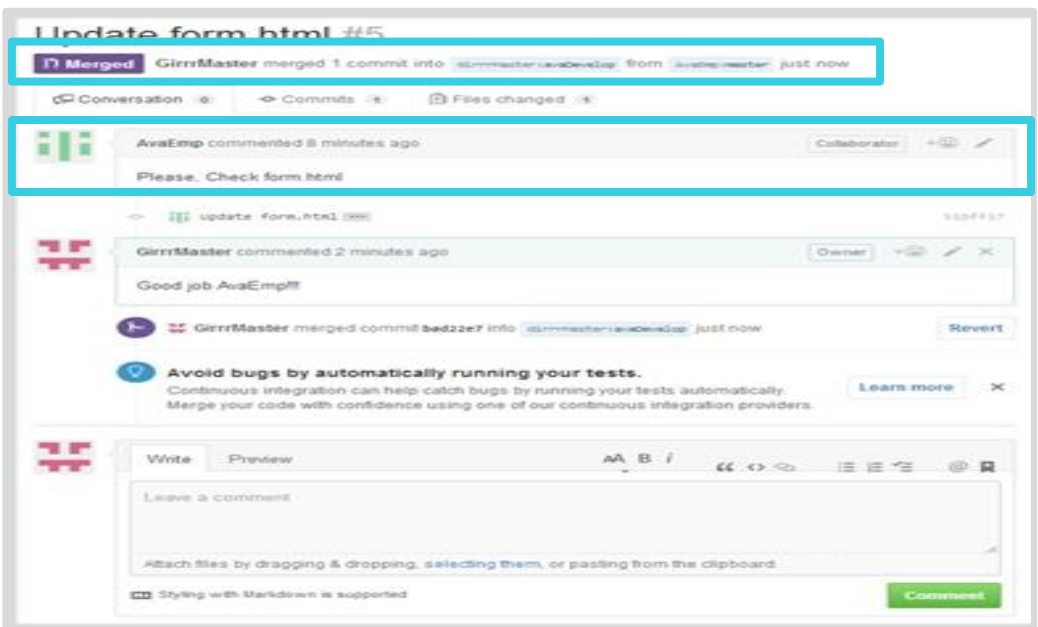
GitHub에서 Code Review 하기

1. Code Review 준비

4) AvaEmp 계정에서 소스코드 업데이트

- merge 메시지 입력 후 , Confirm merge 수행



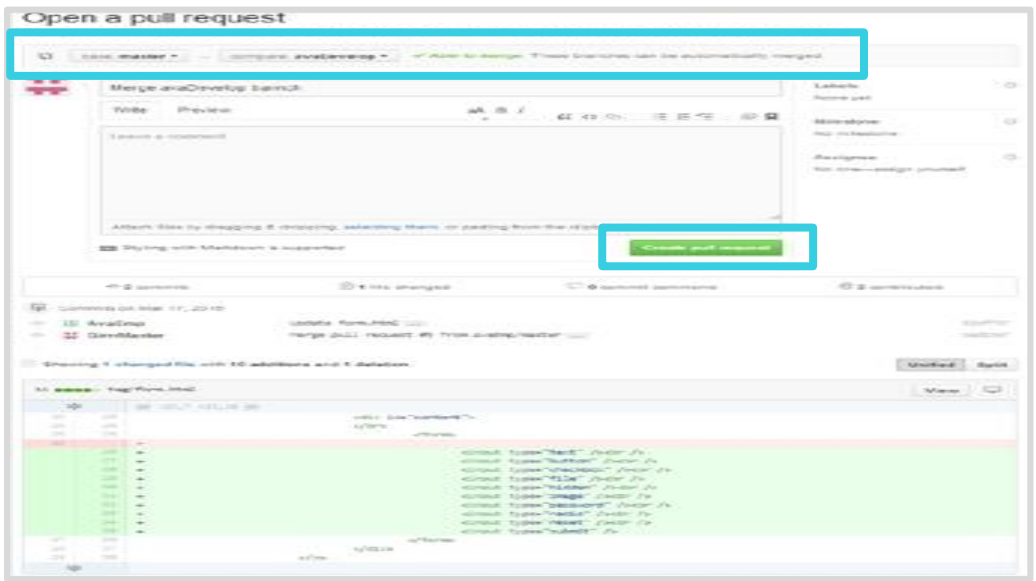


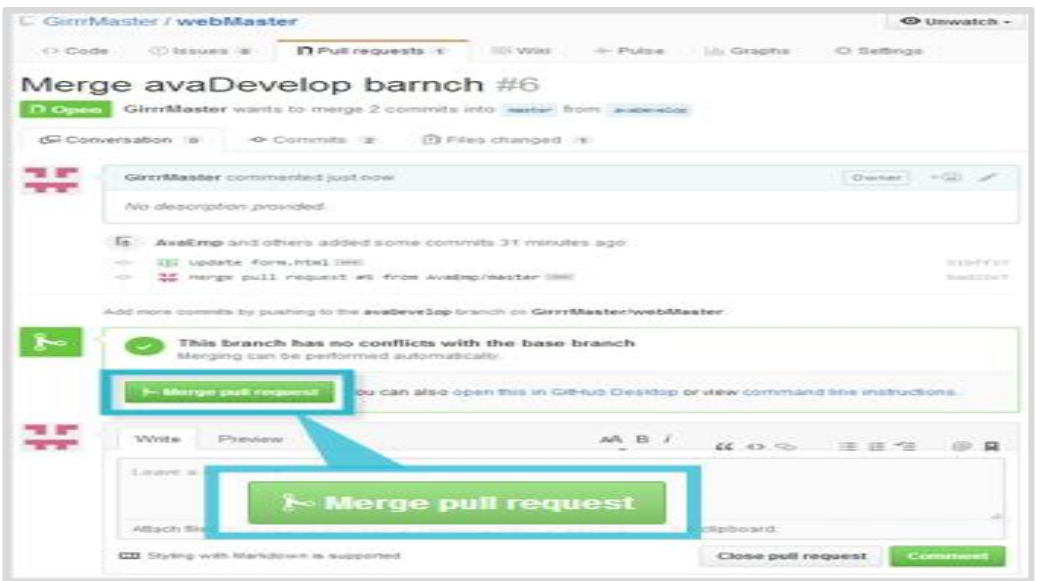
GitHub에서 Code Review 하기

1. Code Review 준비

4) AvaEmp 계정에서 소스코드 업데이트

- pull request 기능을 이용하여 merge 수행



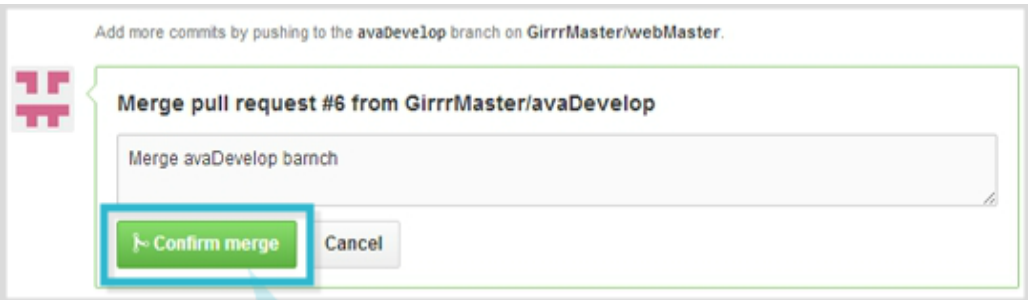


GitHub에서 Code Review 하기

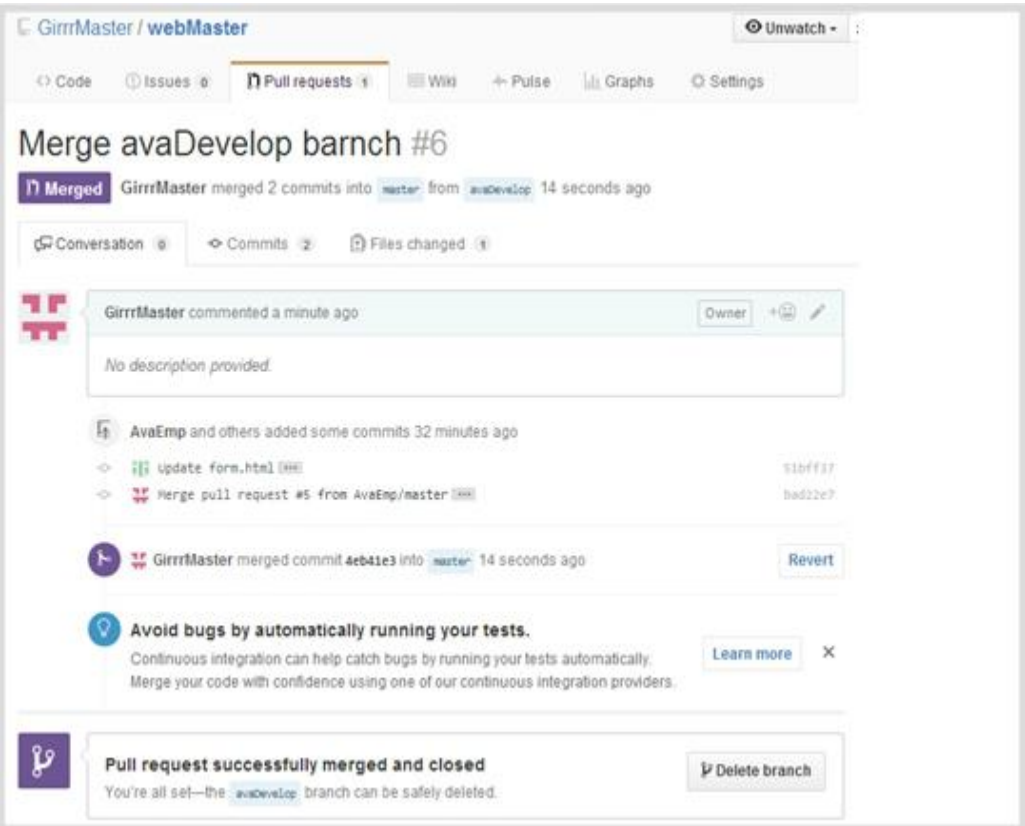
1. Code Review 준비

4) AvaEmp 계정에서 소스코드 업데이트

- pull request 기능을 이용하여 merge 수행



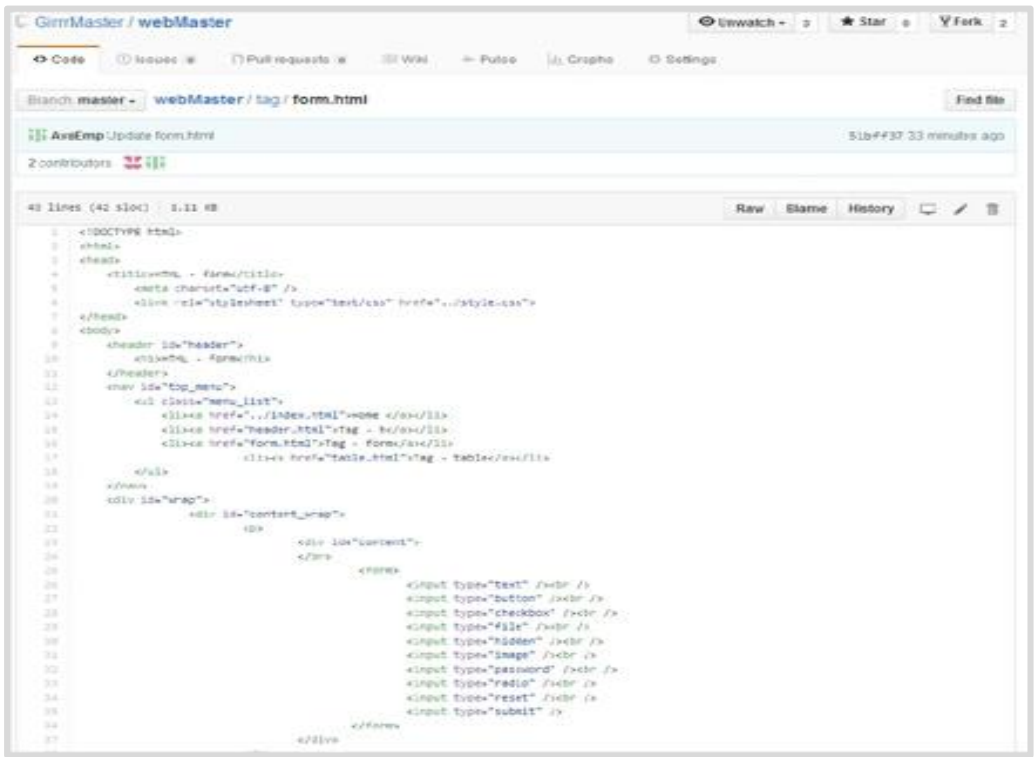
- merge 수행 후 form.html 파일 내용 확인



1. Code Review 준비

4) AvaEmp 계정에서 소스코드 업데이트

- merge 수행 후 form.html 파일 내용 확인

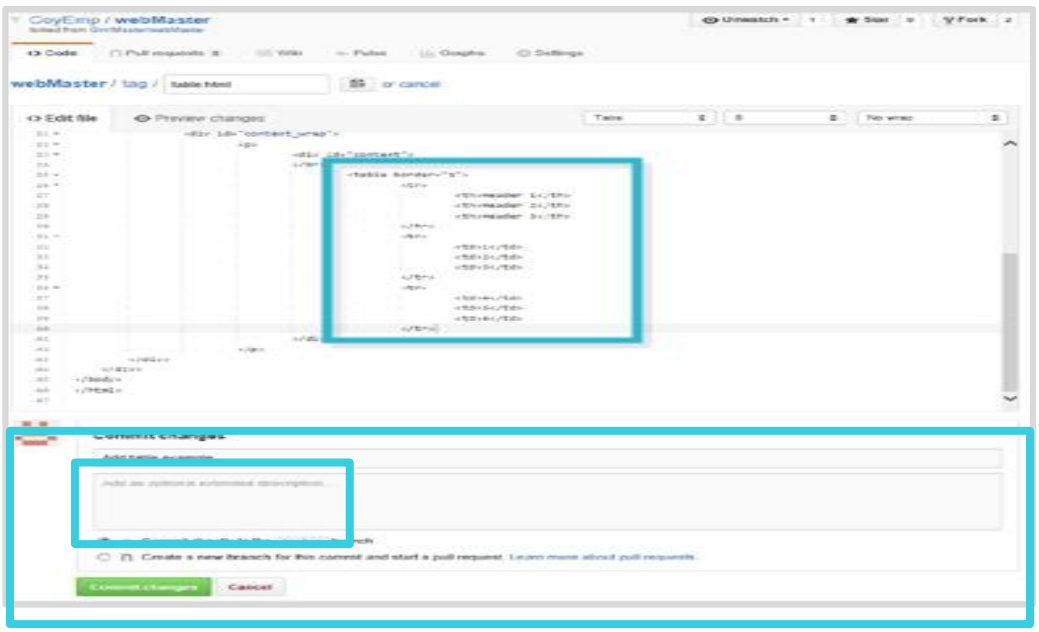


- avaEmp가 직접 master branch로 pull request를 하지 않은 이유
  - 소프트웨어를 개발할 때 동일한 소스코드를 함께 공유하면 여러 사람들이 다양한 버전의 소스코드를 만들어 낼 수 있음
  - 동시에 하나의 소스코드로 작업하게 되면 추후 통합할 때 버그가 발생하기 쉬움
  - 여러 branch를 가지고 서로 다른 버전을 개발하여 나중에 원래의 버전과 비교해서 하나의 새로운 버전으로 만들어 낼 수 있음

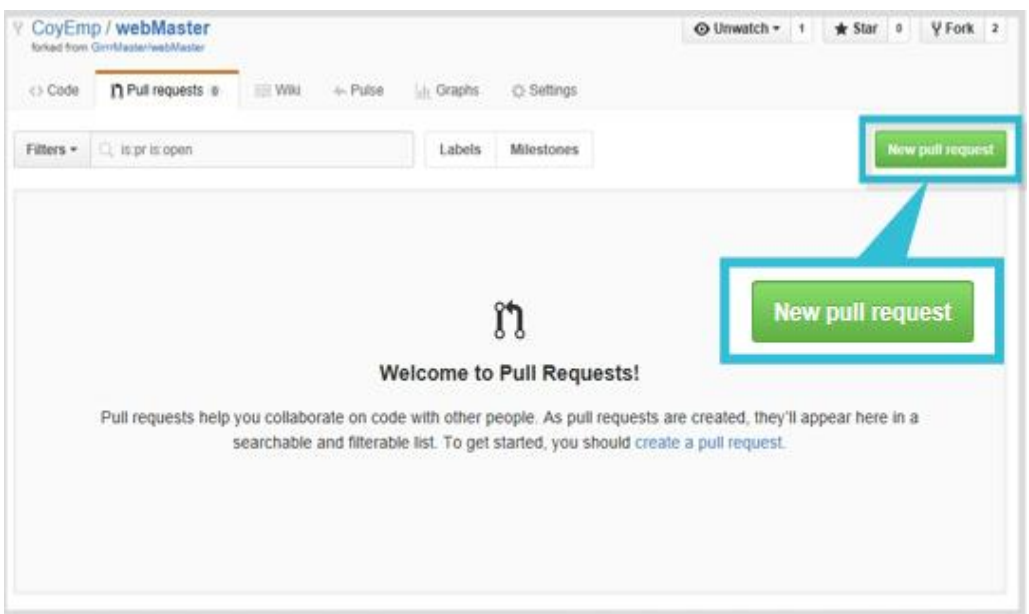
2. Code Review

1) Code Review

- CoyEmp 개발자2 계정에서 table.html 파일 업데이트



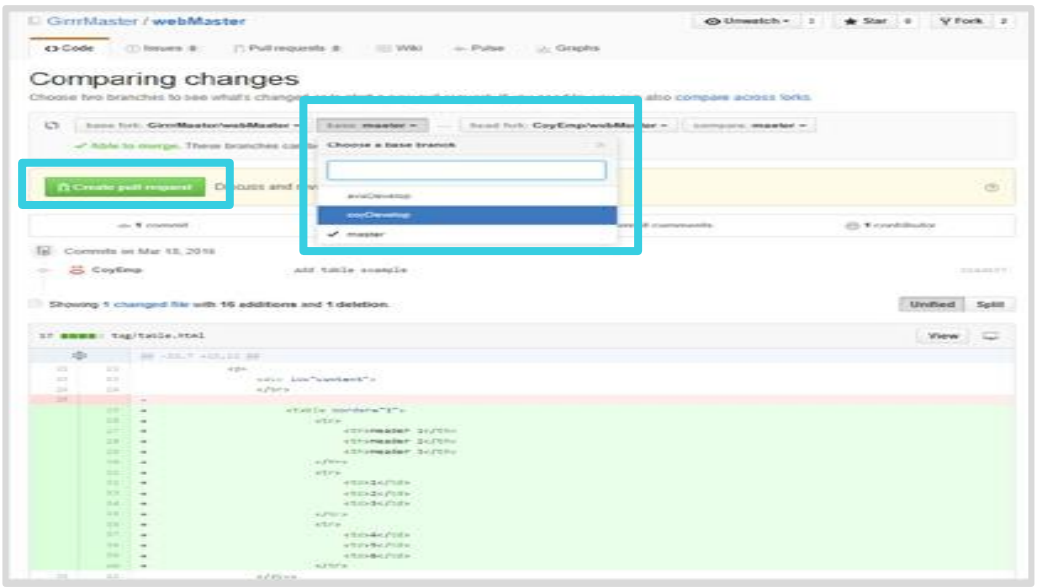
- 업데이트 내용을 girrrMaster에게 pull request 요청

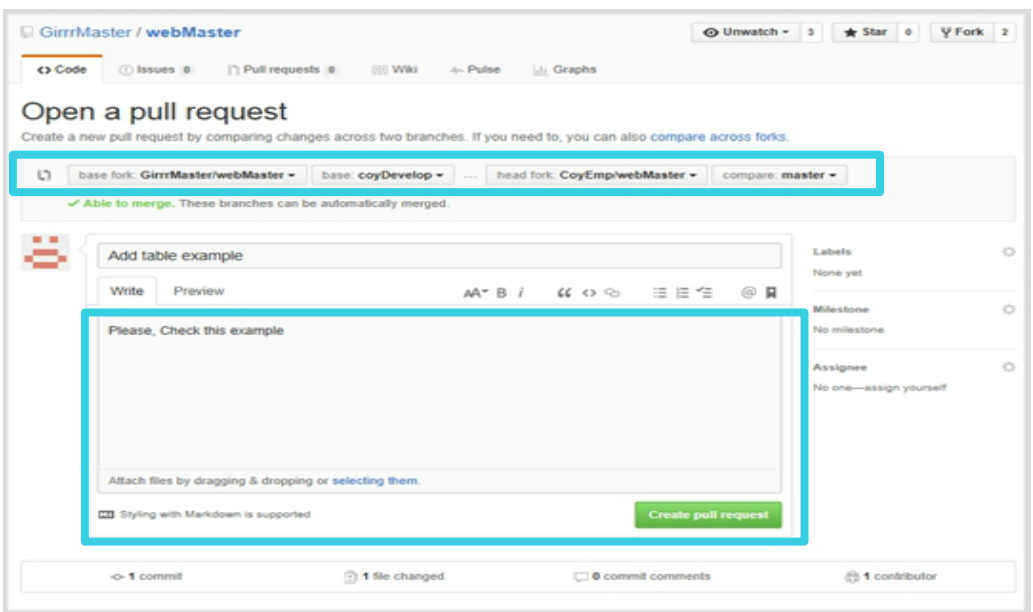


2. Code Review

1) Code Review

- coyDevelop branch로 pull request 요청하기 위해 선택 후 메시지 입력 후 실행

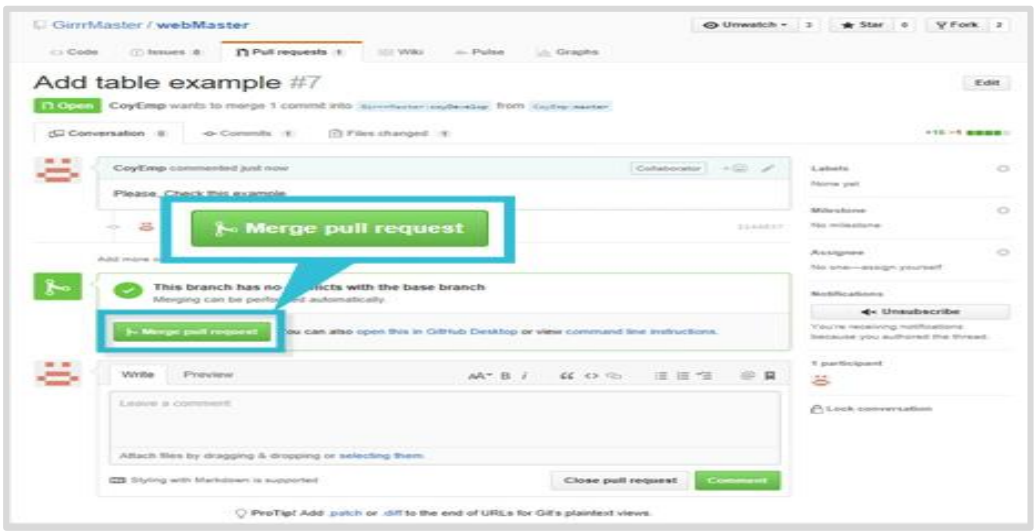




2. Code Review

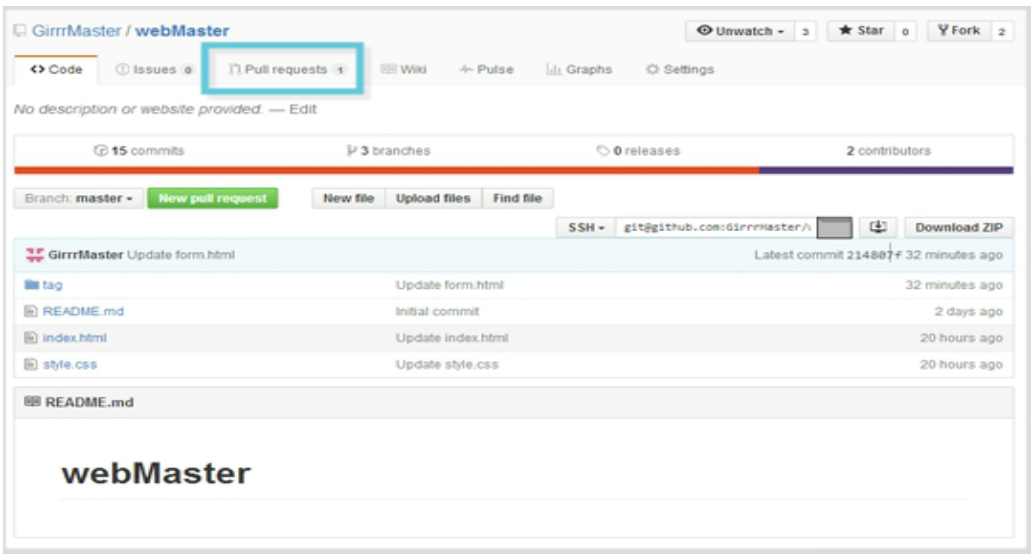
1) Code Review

- merge pull request 버튼 클릭 하고 요청 완료



- pull request와 merge하는 방법과 동일

- girrrMaster 계정에서 pull request 요청을 pull request 탭에서 확인 함

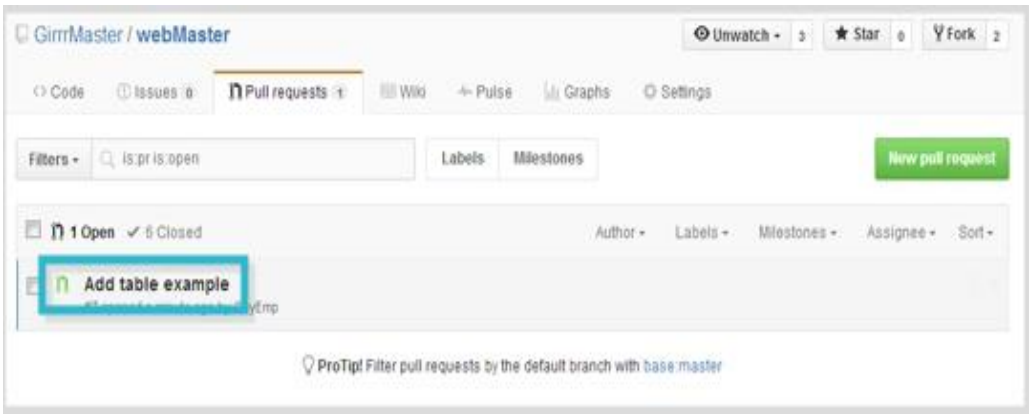


- pull requset 리스트 확인

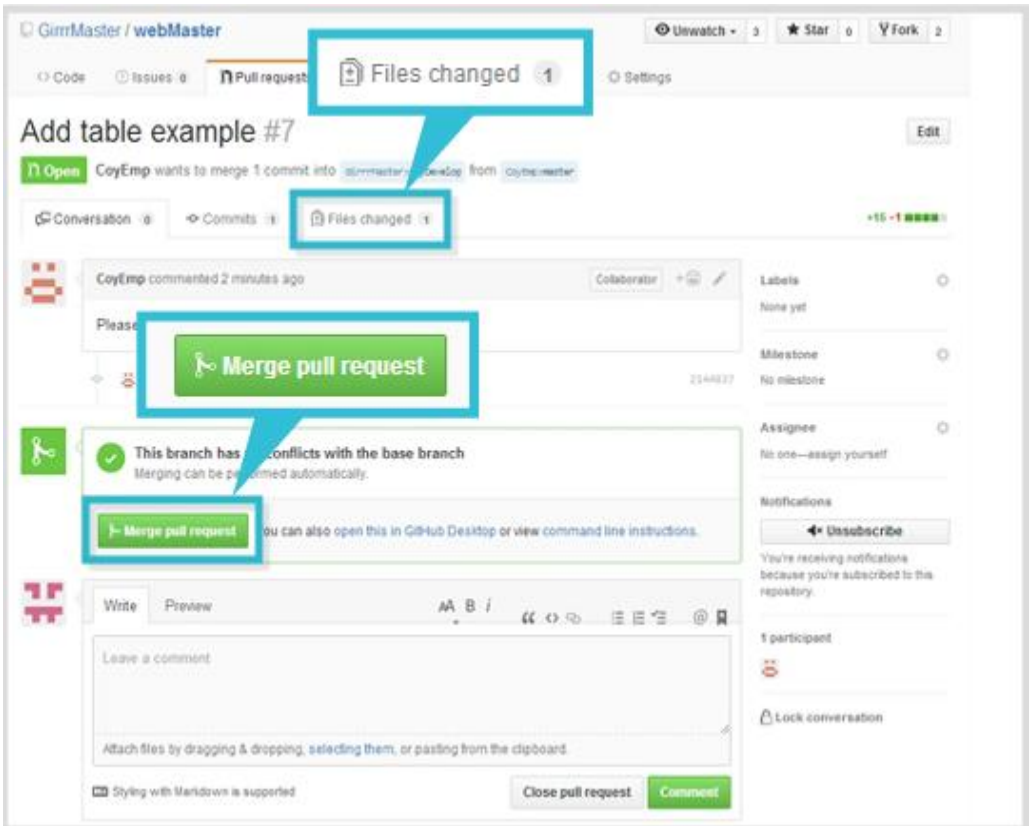
2. Code Review

1) Code Review

- Add table example pull request 클릭



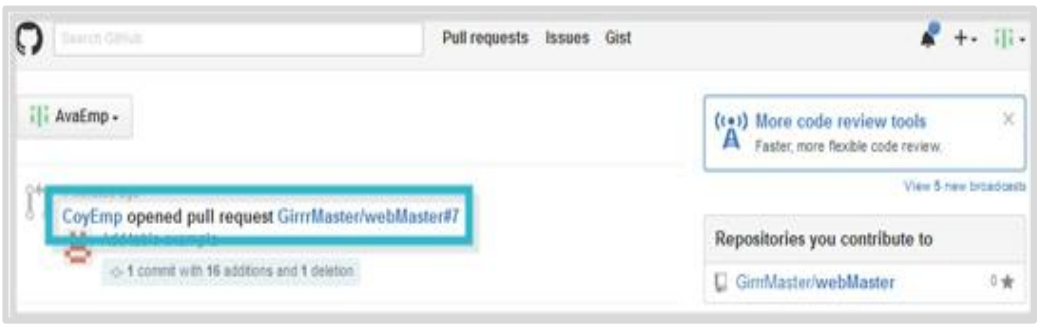
- Files changed에서 table.html 파일의 수정 부분 확인 가능



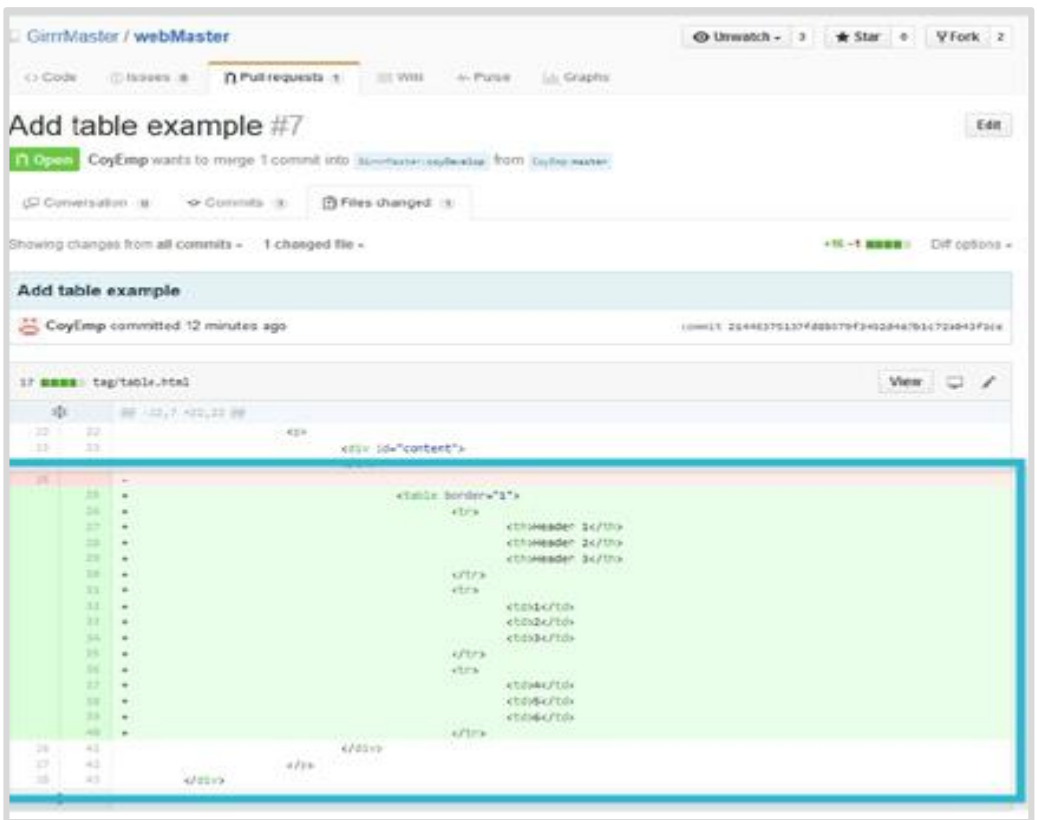
2. Code Review

2) 파일 내용 review

- 개발자1 계정에서 pull request 내용으로 이동



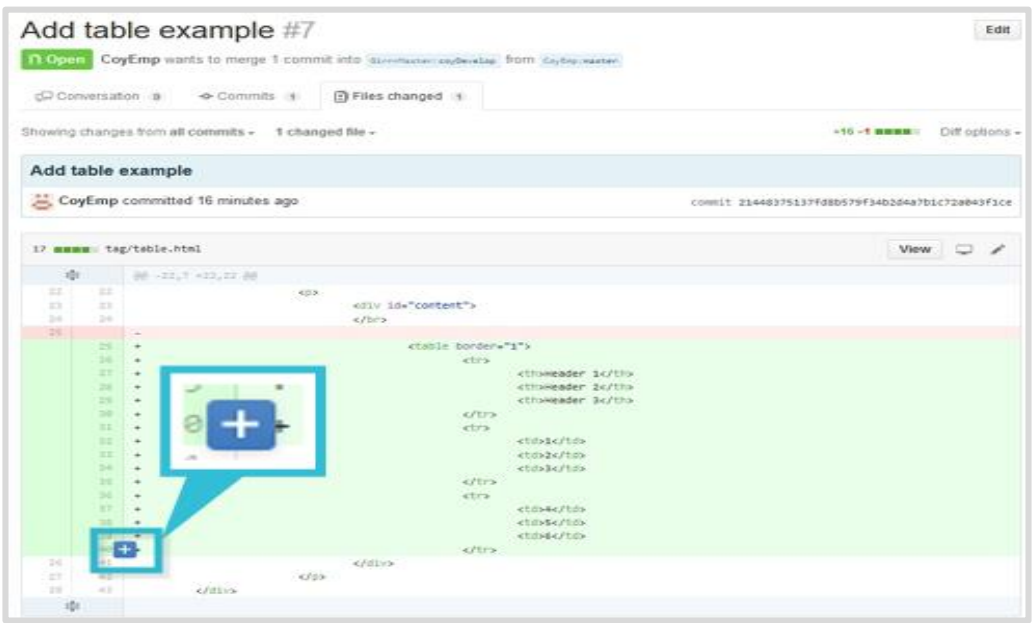
- coyEmp 개발자 2가 수정한 파일 내용 확인 가능



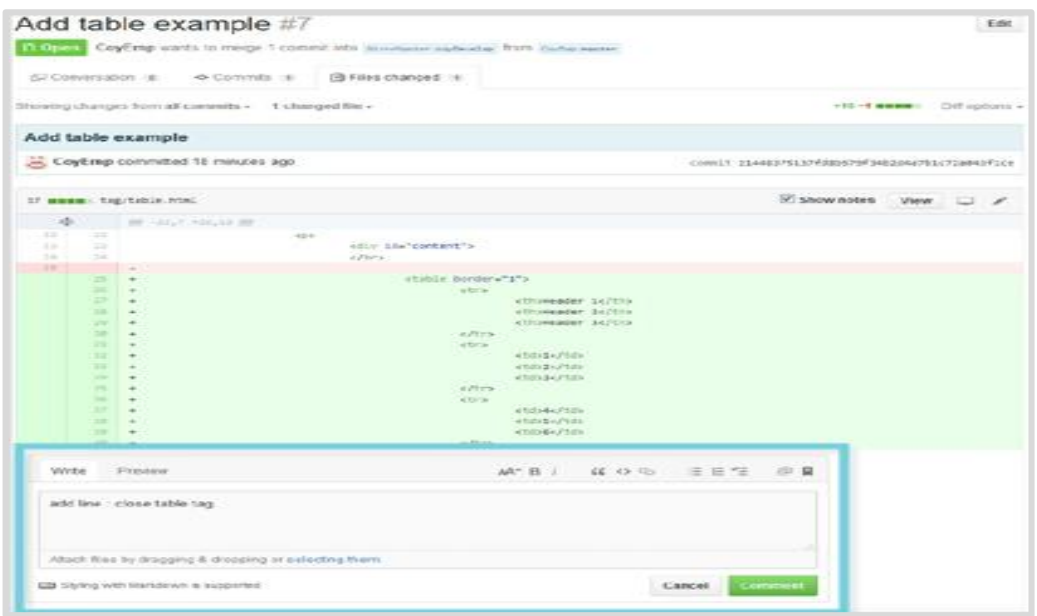
2. Code Review

2) 파일 내용 review

- 수정된 파일 내용의 '+' 표시된 부분을 클릭하면 comment 입력창 생성



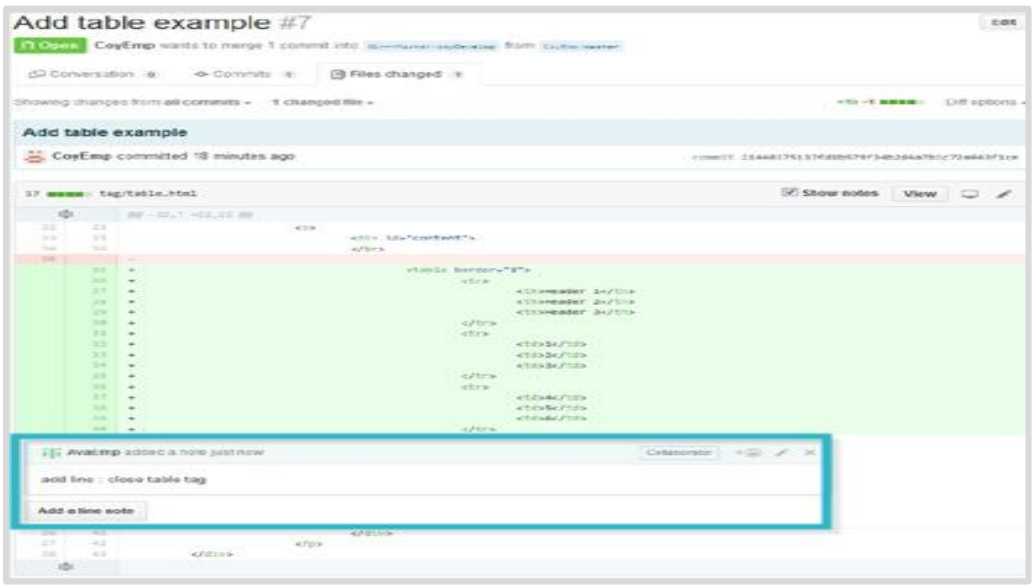
- 오류수정을 위한 Code Review



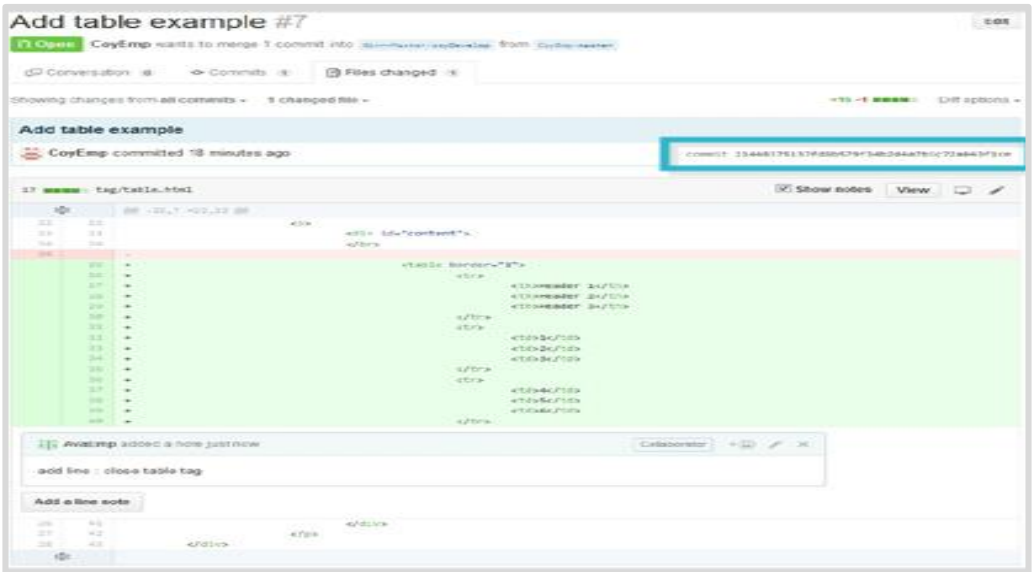
2. Code Review

2) 파일 내용 review

- 오류수정을 위한 Code Review



- commit 번호를 복사하여 저장소에서 Issue를 생성하여 Code Review 내용 등록

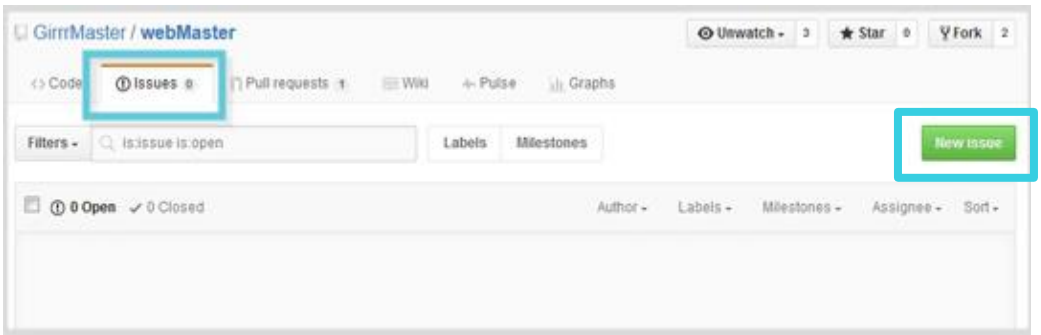


- 저장소의 소유자인에게 issue를 보내기 위해서

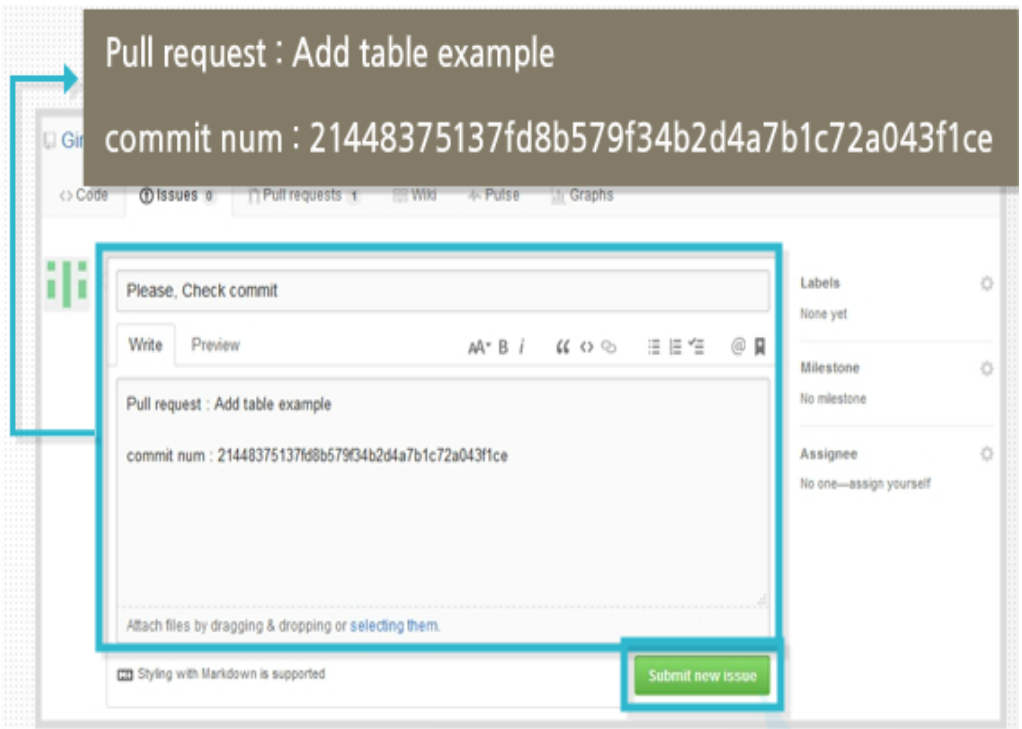
2. Code Review

2) 파일 내용 review

- commit 번호를 복사하여 저장소에서 Issue를 생성하여 Code Review 내용 등록



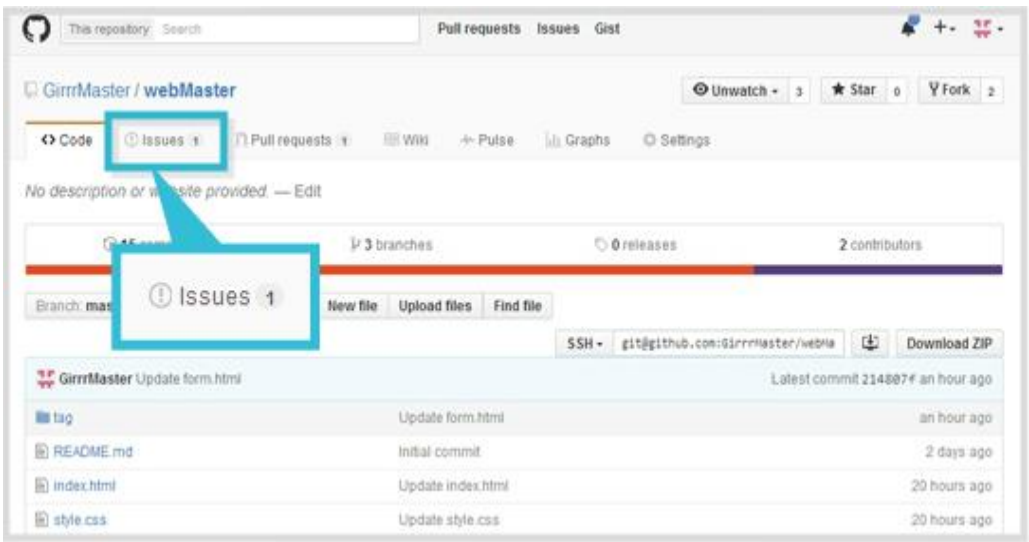
- Submit New issue 버튼을 이용하여 issue 생성 완료



2. Code Review

2) 파일 내용 review

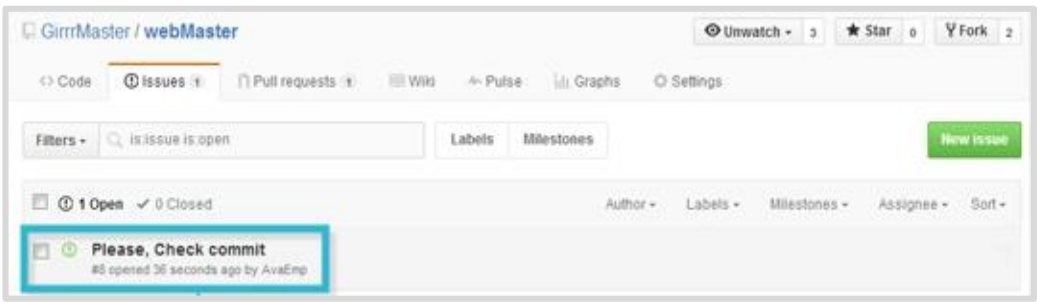
- Submit New issue 버튼을 이용하여 issue 생성 완료



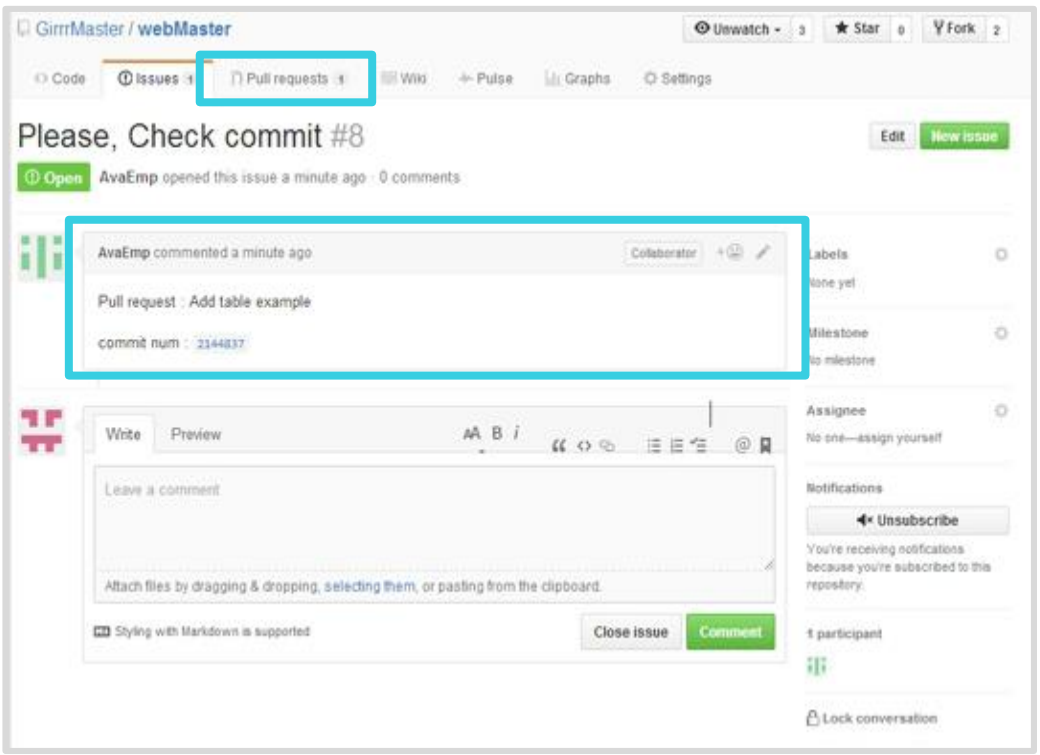
2. Code Review

3) Issue 생성 및 수정

- girrrMaster 계정에서 해당 issue로 이동



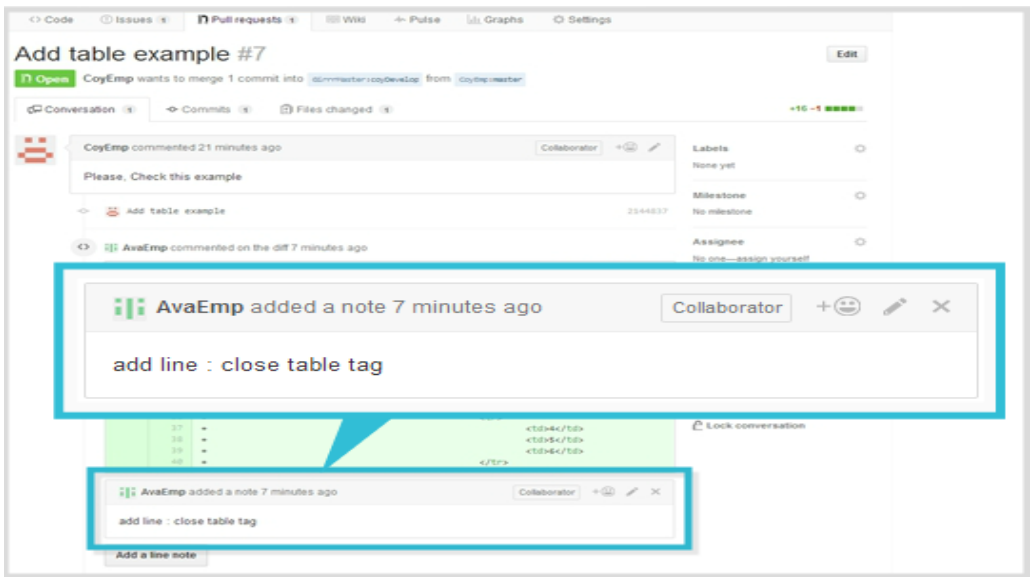
- issue 내용에 맞게 해당 Code Review로 이동



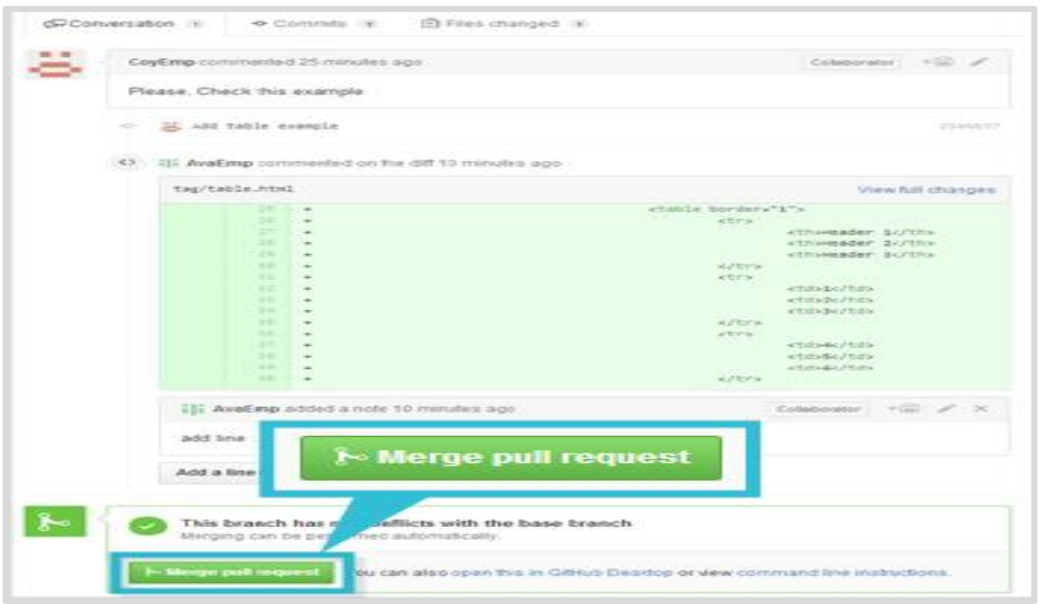
2. Code Review

3) Issue 생성 및 수정

- issue 내용에 맞게 해당 Code Review로 이동



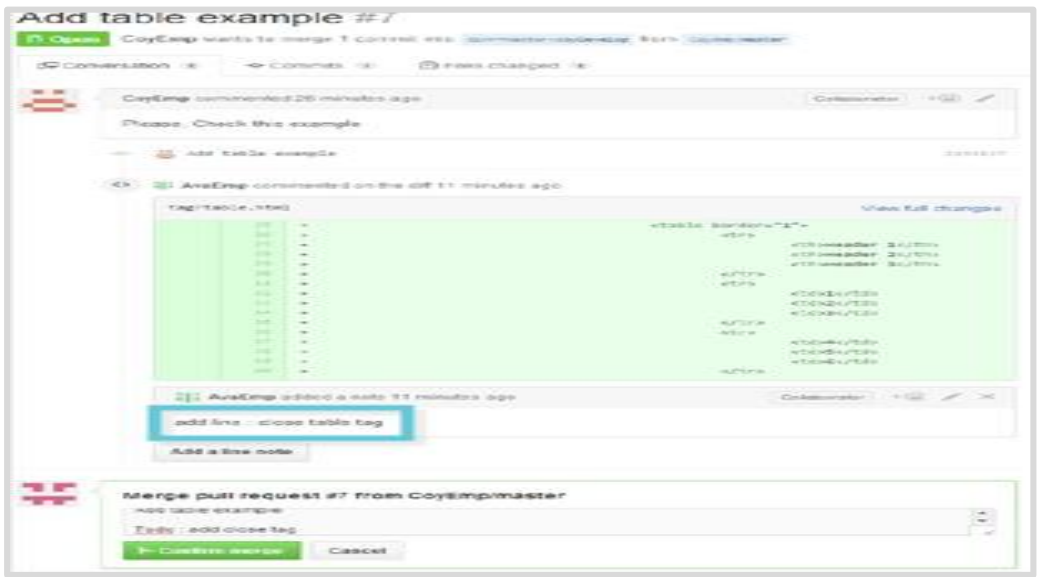
- Merge pull request 클릭 후 comment를 입력하고 소스코드 반영



2. Code Review

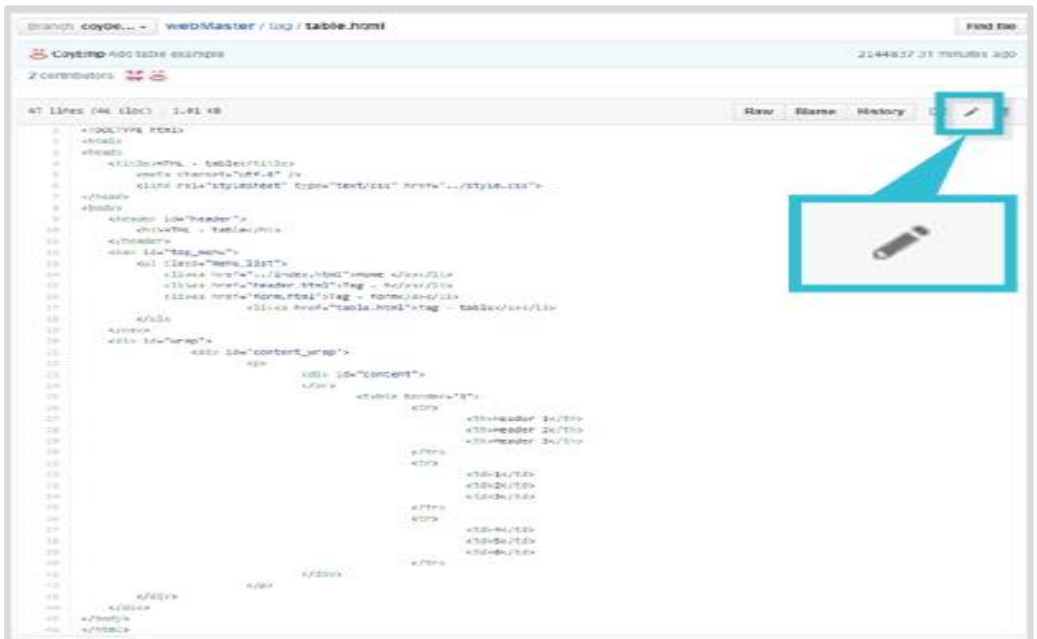
3) Issue 생성 및 수정

- Merge pull request 클릭 후 comment를 입력하고 소스코드 반영



- merge 메시지에 수정하여 반영

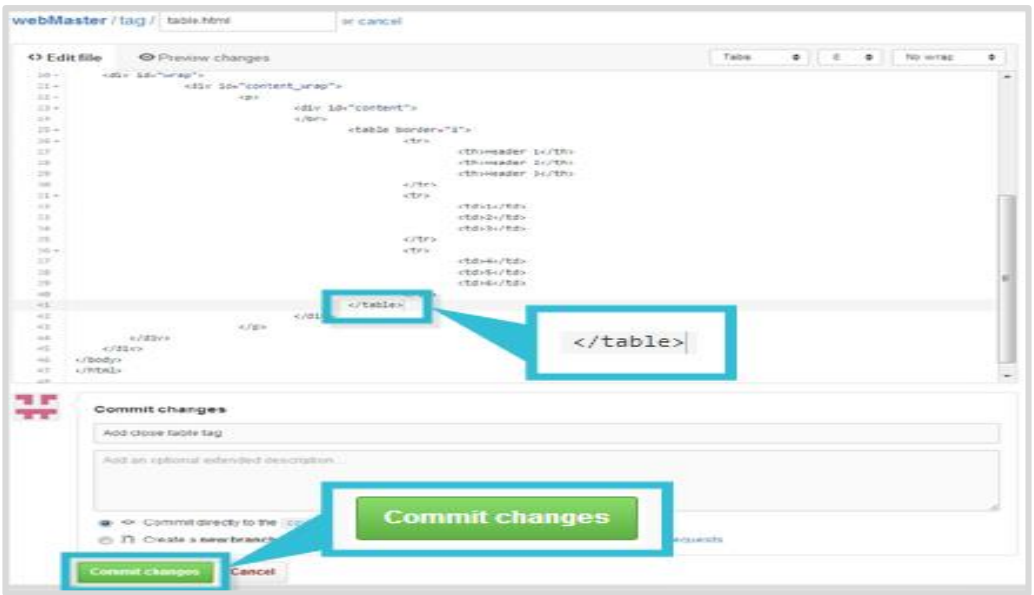
- 파일에서 issue로 등록된 내용 수정 후 commit



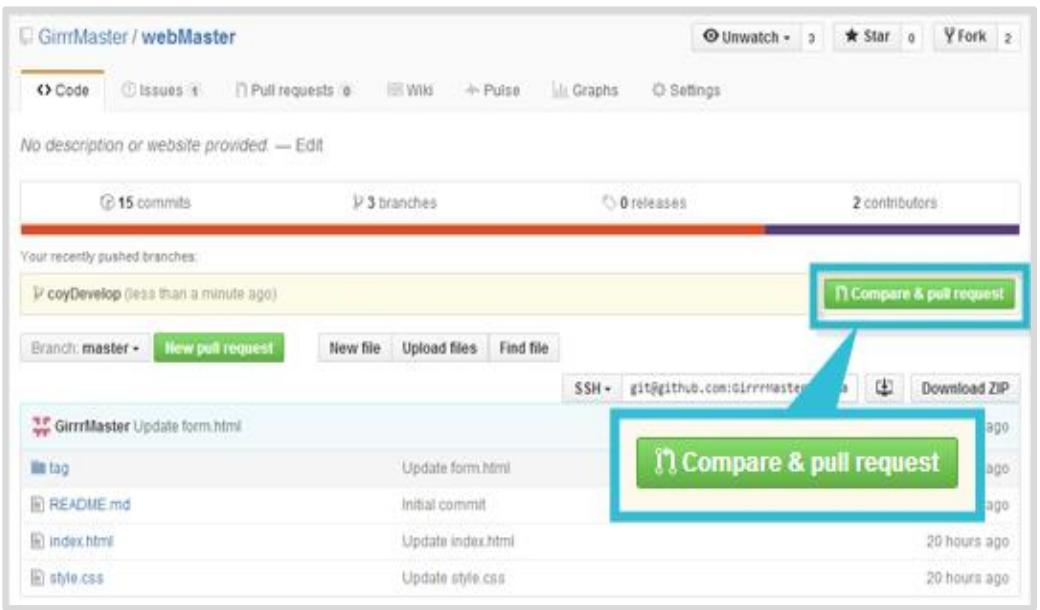
2. Code Review

3) Issue 생성 및 수정

- 파일에서 issue로 등록된 내용 수정 후 commit



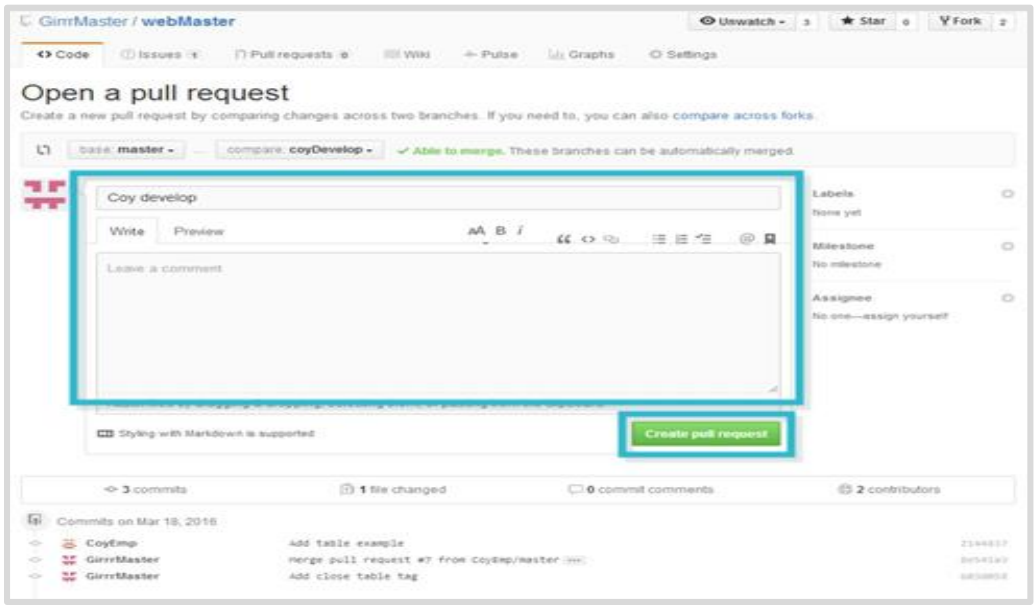
- master branch로 업데이트 된 파일 merge



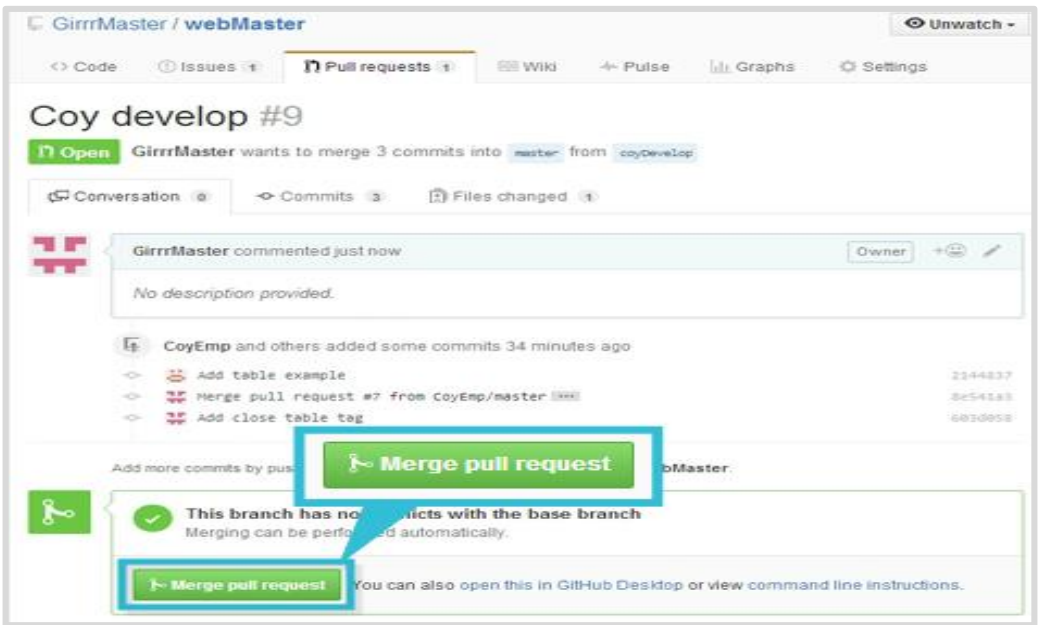
2. Code Review

3) Issue 생성 및 수정

- branch 간에 합병을 위한 pull request 수행



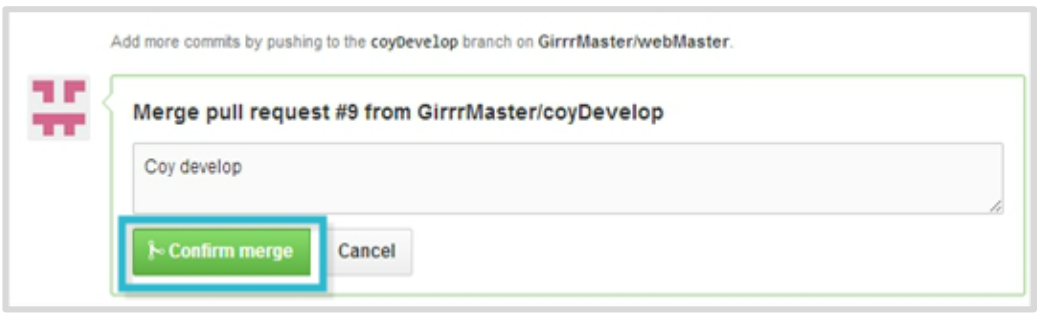
- Merge pull request 기능으로 합병

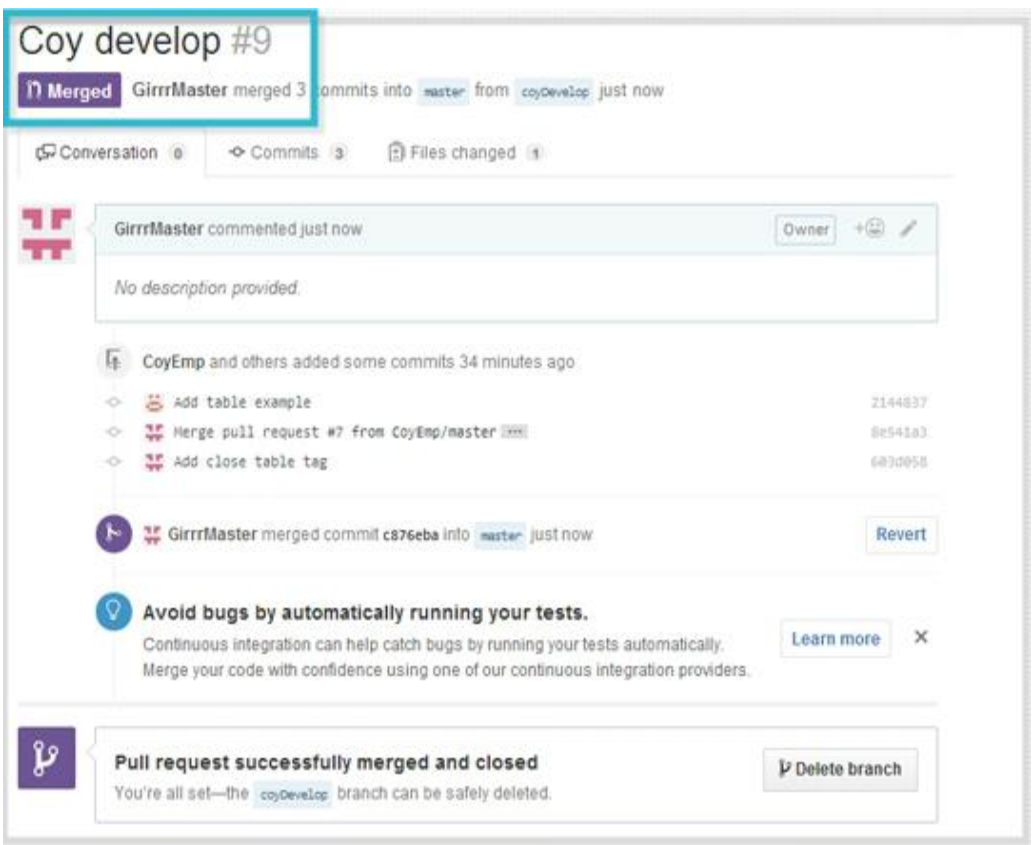


2. Code Review

3) Issue 생성 및 수정

- merge comment를 입력하고 Confirm merge 클릭하여 merge 완료

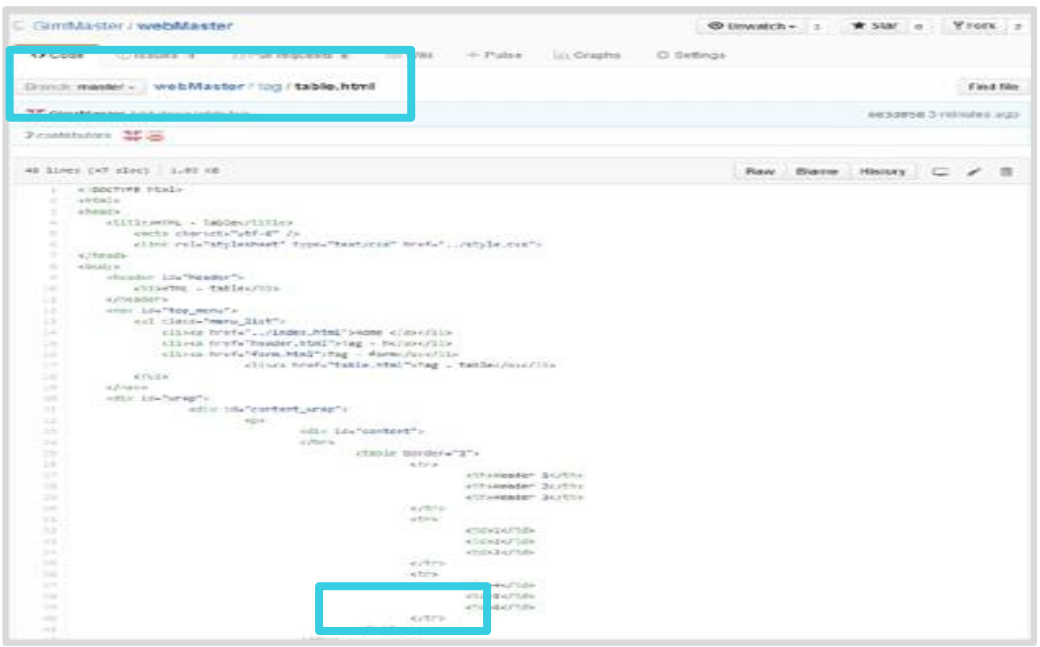




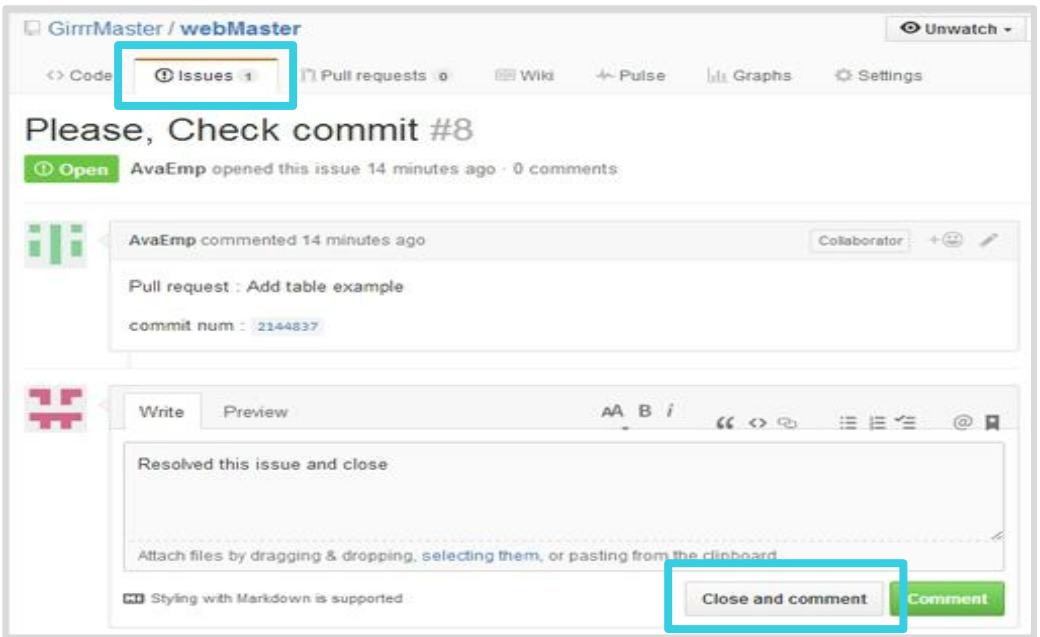
2. Code Review

3) Issue 생성 및 수정

- Master branch에 merge된 table.html 파일 확인



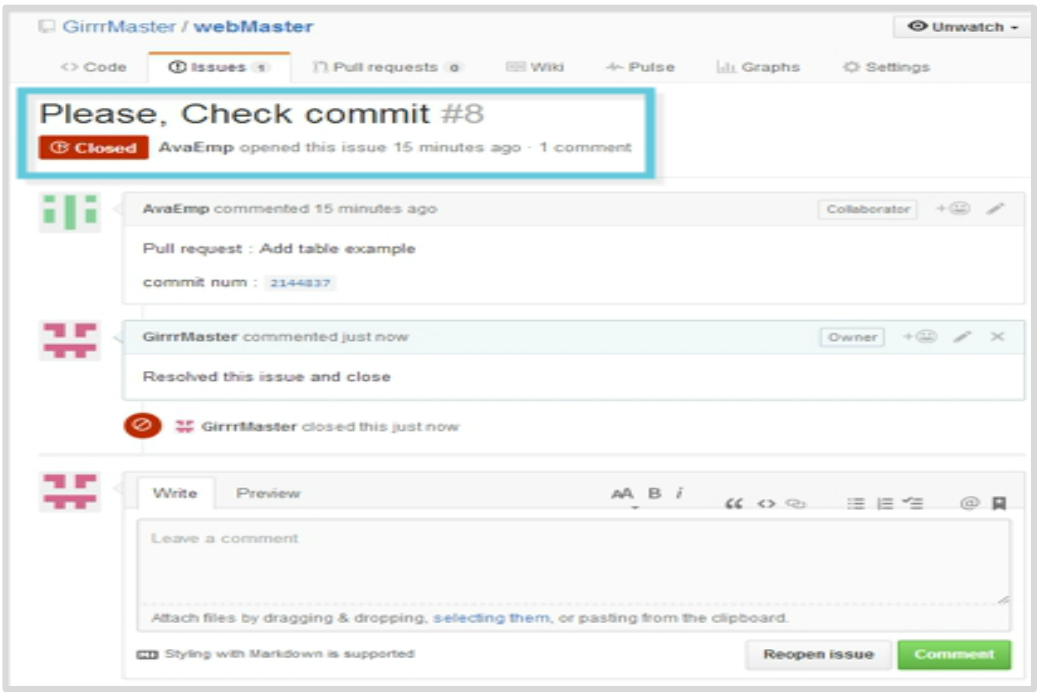
- issue를 해결완료하고 issue Close



2. Code Review

3) Issue 생성 및 수정

- issue를 해결완료하고 issue Close



핵심정리

GitHub에서 Code Review 하기

1. Code Review 준비

- 로컬저장소 및 파일 생성 → 로컬저장소에서 Branch 생성 → Fork 기능 수행 및 Merge 완료 → AvaEmp 계정에서 소스코드 업데이트

2. Code Review 수행

- coyEmp 개발자 계정에서 table.html 파일을 업데이트 → 업데이트 내용을 girrrMaster에게 pull request 요청 → coyDevelop branch로 pull request 요청하기 위해 선택 후 메시지 입력 후 실행 → merge pull request 버튼 클릭 하고 요청 완료 → GirrrMaster 계정에서 pull request 요청을 pull request 탭에서 확인 → Add table example pull request 클릭 → Files changed에서 table.html 파일의 수정 부분 확인 가능

핵심정리

Code Review 실습

1. 소스코드 생성 및 반영

- 소스코드 생성
- 소스코드 서버에 반영

2. Code Review 등록

- [Code] – [commits]에서 최신의 commit 확인
- Code에 + 버튼을 클릭하여 Review Comment 작성
- Code Review 확인 및 추가

3. Code Review 편집

- 수정 아이콘을 클릭하여 추가 및 수정
- X 버튼을 클릭하여 삭제