

# github & github desktop 사용법

과목명 : 인터넷 활용

교수님 : 이동은 교수님

학 과 : 컴퓨터 공학과

학 번 : 20180360

이 름 : 박현정

# <github 사용법>

## 1. github 계정 만들기

## 2. 저장소 생성하기

- start a project -> create repository 입력 후 아래 빈칸 입력
- repository name : 저장소의 이름, 사용자가 원하는 이름 입력
- public, private : 무료 결제시 public 이용해서 여러명이 공유

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

The screenshot shows the GitHub 'Create a new repository' form. At the top, there are two input fields: 'Owner' with a dropdown menu showing 'parkhyeonjeong0429' and 'Repository name' with a text input containing 'testweb' and a green checkmark. Below these is a hint: 'Great repository names are short and memorable. Need inspiration? How about supreme-palm-tree?'. The 'Description (optional)' field contains the text '과제 하기'. Under the 'Visibility' section, 'Public' is selected with a radio button, and 'Private' is unselected. Below this, there is a checkbox for 'Initialize this repository with a README' which is checked. At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None', followed by a green 'Create repository' button.

## 3. 새 분기 만들기

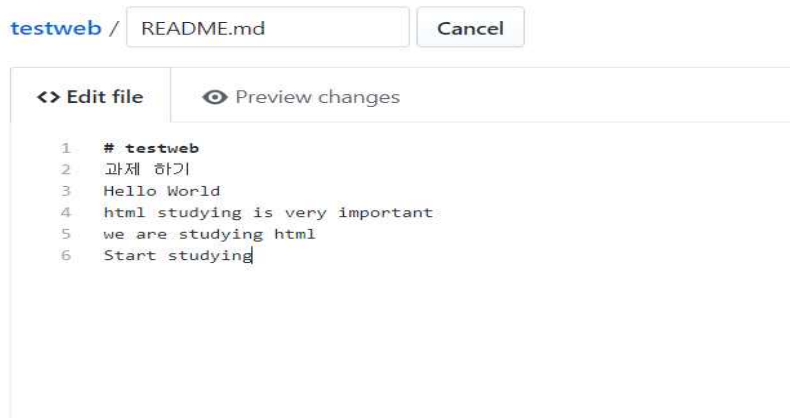
“Branch:master” 선택 후 “html\_test” 새로운 분기 이름 추가한다.

The screenshot shows the GitHub 'Switch branches/tags' dropdown menu. At the top, there is a 'Branch: master' dropdown and a 'New pull request' button. The dropdown menu is open, showing a search bar with 'html\_test' entered. Below the search bar, there are two tabs: 'Branches' and 'Tags'. Under the 'Branches' tab, there is a blue icon and the text 'Create branch: html\_test from master'.

## 4. 변경사항 만들기 및 커밋

REMADE.md 클릭 -> testweb에 편집하기 버튼 누른다. -> 에디터에 자신에 대해 적기 -> 작성후 commit change 클릭( 이변경 사항은

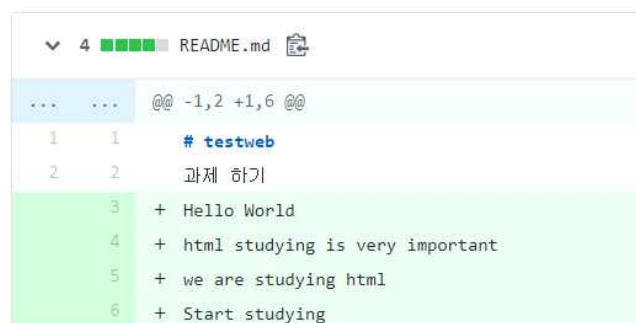
html\_test 지점의 REMADE 파일에만 적용된다.



## 5. 끌어오기 & 요청 열기

pull requests -> New pull request 누른 후 아까 만든 html\_test파일 확인하기

Showing 1 changed file with 4 additions and 0 deletions.



-> (+) 기호에 있는 것들이 내가 따로 추가한 내용이다.  
(이 변경 사항들이 맞다면 create pull request 버튼)

## 6. 끌어오기 & 요청병합

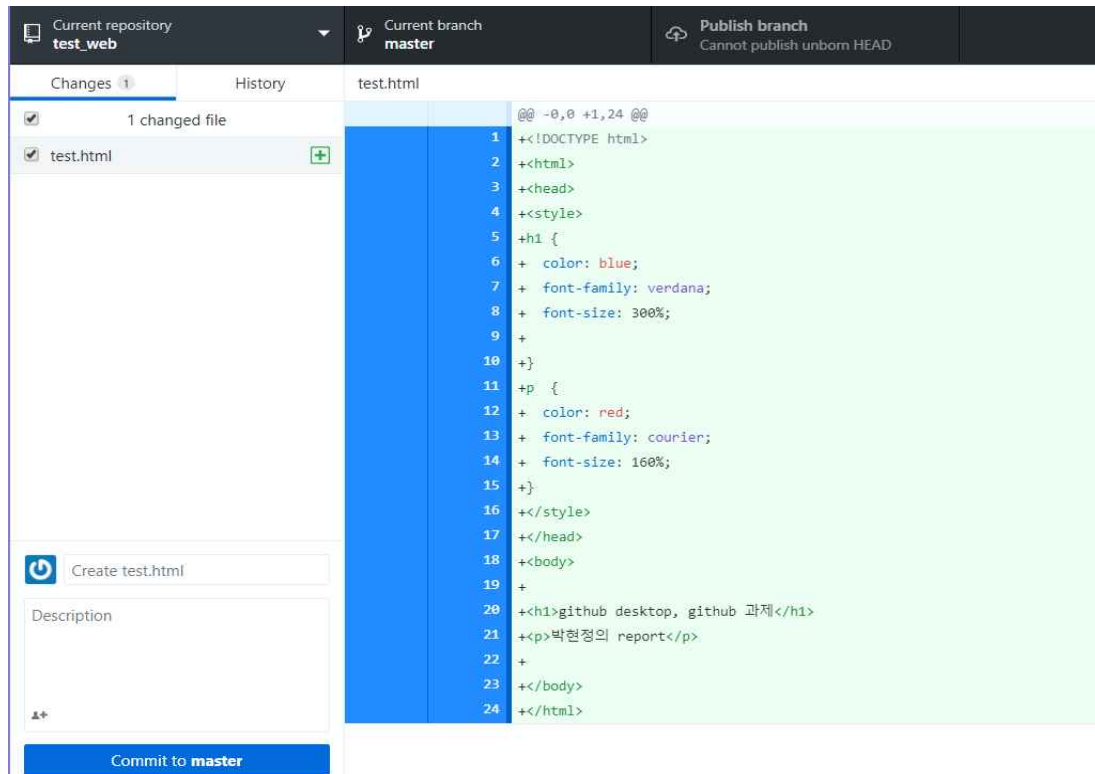
merge pull request 눌러서 변경사항을 병합한다.

# <github desktop으로 홈페이지 만들기>

github desktop은 홈페이지를 더욱 편리하게 만들 수 있도록 도와준다.

1. github에서 github desktop에서 소스들을 관리하기 위해 new repository를 만들어 준다.
2. 여기서 생성되는 url을 복사해서 github desktop에서 clone repository에 url 입력하여 장소 선택하여 만들어 준다.(만들어준 repository가 프로그램과 연동된다.)

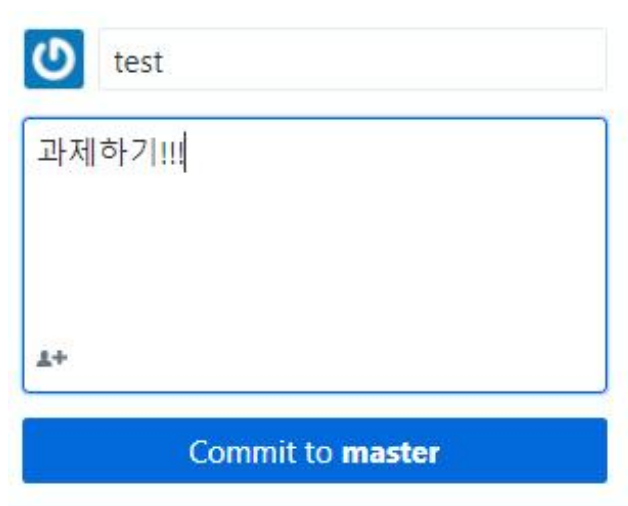
3. html문을 작성하여 아까 만들어진 "test\_web" 파일에 넣는다.



-> 그러면 이렇게 오른쪽에 입력한 html 문서가 나타나게 된다.

4. 그 후 push를 입력하여 github 웹 사이트에도 등록한다.

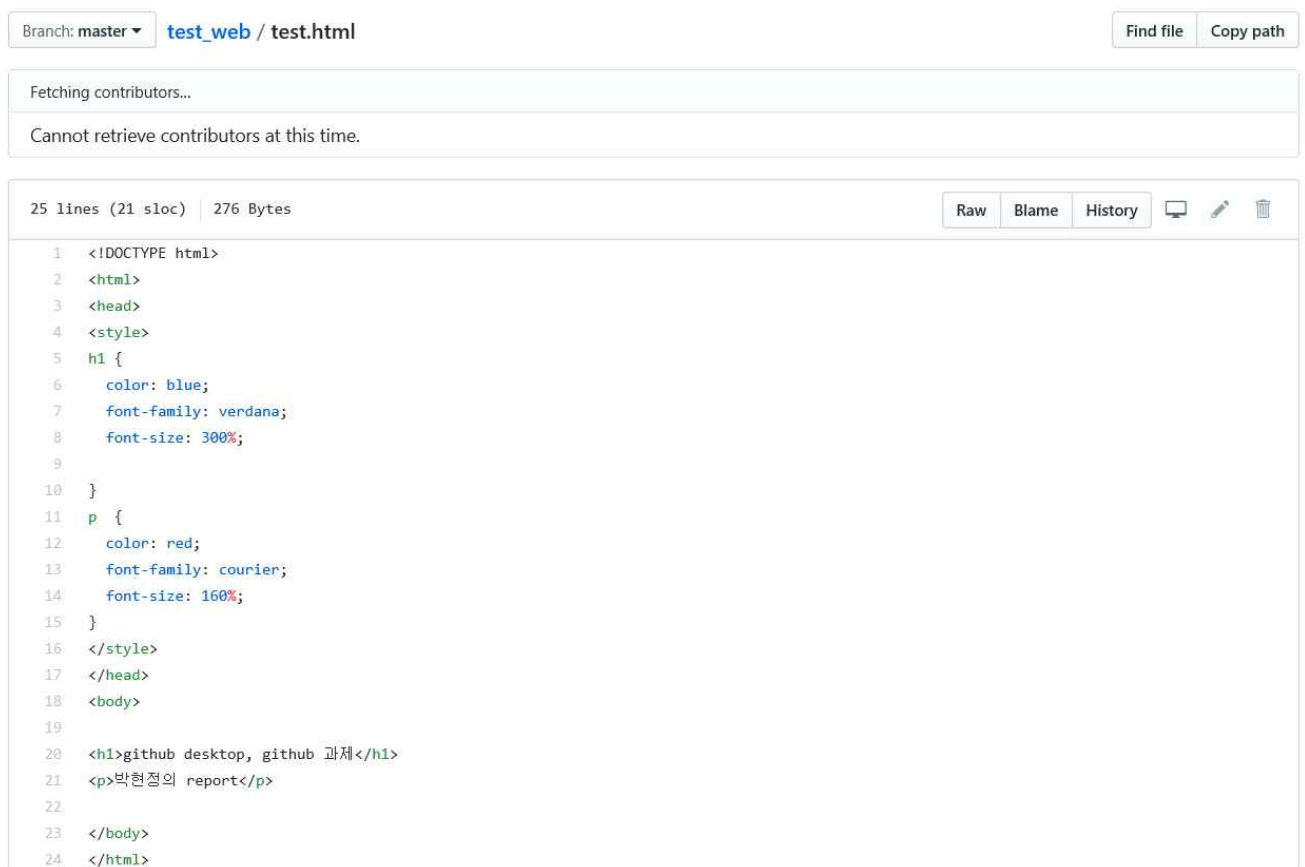
5. summary와 description 또한 모두 저장해준다.



6. github 웹사이트에 들어가면 아까 만든 test\_web에 test.html 파일이 들어 있는 것을 확인 할 수 있다.

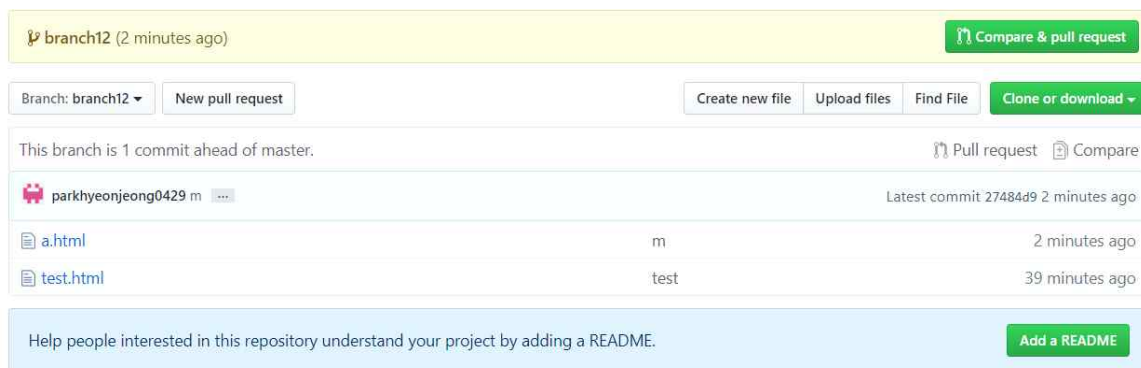


<아래 사진은 github에 등록된 test.html 파일이다.>



7. branch 라는 기능을 사용 할 수 있다.

- master가 아닌 다른 이름으로 사용 할 수 있으며 다른 사람의 원격으로 수정하고 게시 할 수 있고 원래 파일은 그대로 유지된다.



a.txt파일을 barnch12라는 barnch로 저장하면 branch12라는 이름으로 파일이 들어간 것을 확인 할 수 있다.

< 커밋과 푸쉬의 차이 >

- 커밋 : 깃은 소스 관리를 위한 분산 버전 관리 시스템이다. 파일을 수정한다면 그 기록이 깃허브 데스크탑에 남게 된다. 이때 그 기록 저장하는 것이 커밋이다. 커밋(Commit)하기 위한 정보를 왼쪽 아래 Summary와 Description를 작성

- 푸쉬 : 앞에 커밋은 변경 내용을 저장하는 것이다. 변경 내용을 사용자와 공유하기 위하여 서버에 업로드 해야하는데 이것을 푸쉬 라고 한다. Push origin을 클릭하면 서버로 업로드가 됩니다. 이로써 변경 내용이 적용되고 우리는 바뀐 내용을 확인할 수 있게 된다.

<이렇게 하여 github desktop을 이용하여 홈페이지를 더 간단히 만들고 github과 연동시켜서 파일을 저장 할 수 있다.

github과 github desktop을 이용하여 홈페이지를 만들고 관리 할 수 있다.>