

# Fastcampus

## Frontend Dev SCHOOL

Collaborate with git, github



Linux



## Before Linux



- 1965년 데니스 리치, 켄 톰슨 외 x명이 AT&T Bell 연구소에서 PDP-7 기반 어셈블리어로 작성한 UNIX를 개발

## Before Linux



- 1973년 데니스 리치와 켄 톰슨이 C를 개발한 뒤, C 기반 UNIX 재작성

# Before Linux

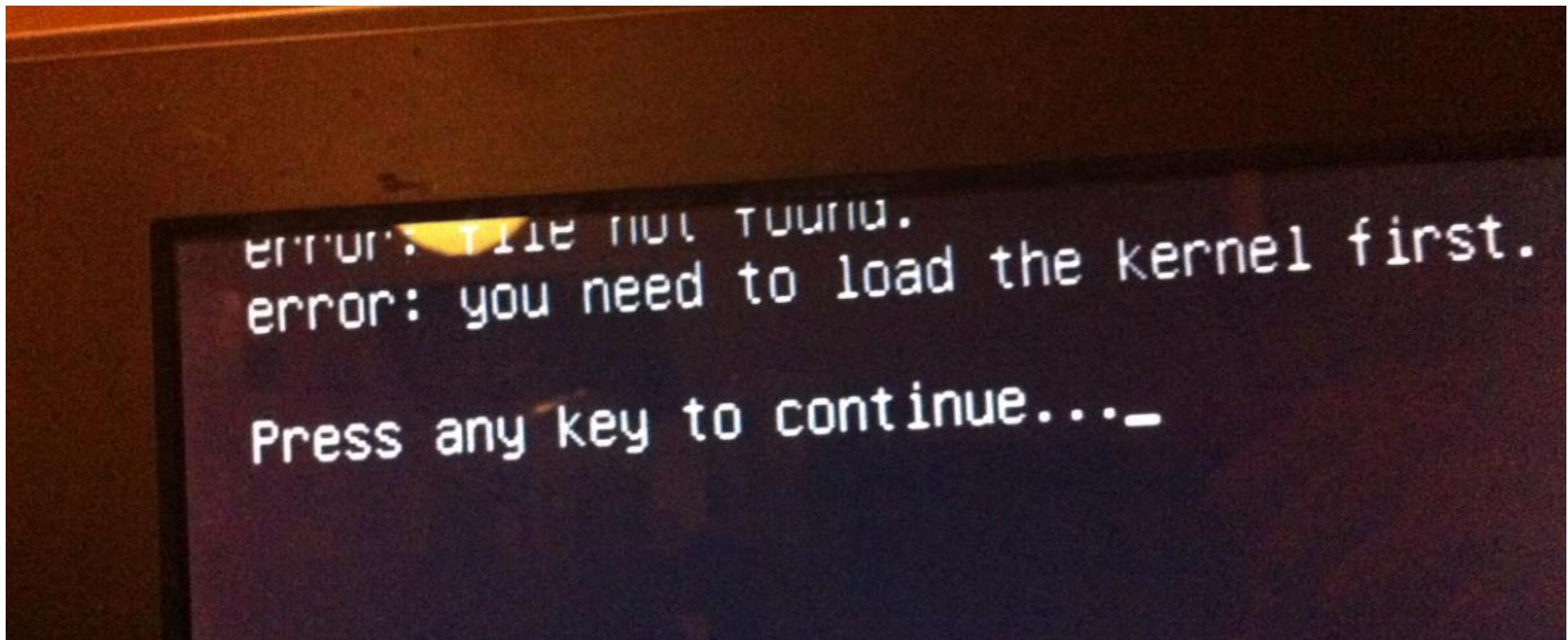


- 1984년 리차드 스톤먼이 오픈 소프트웨어 자유성 확보를 위한 GNU 프로젝트 돌입

## Meaning of GNU

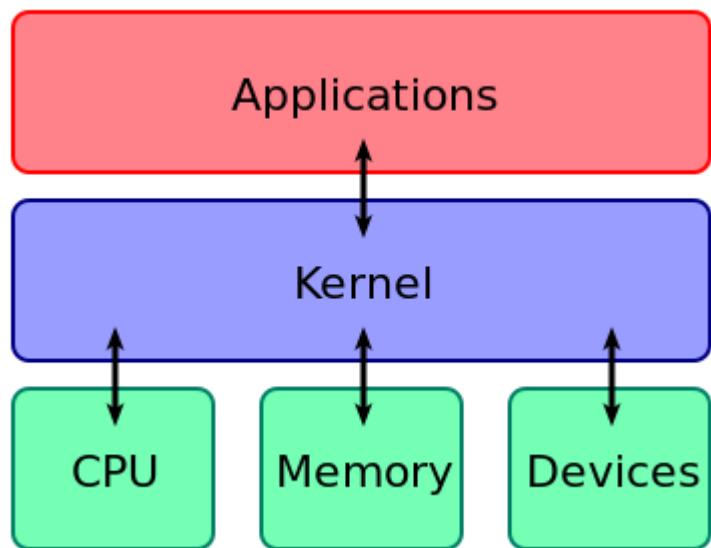
GNU == G NU is N ot U nix

## Before Linux



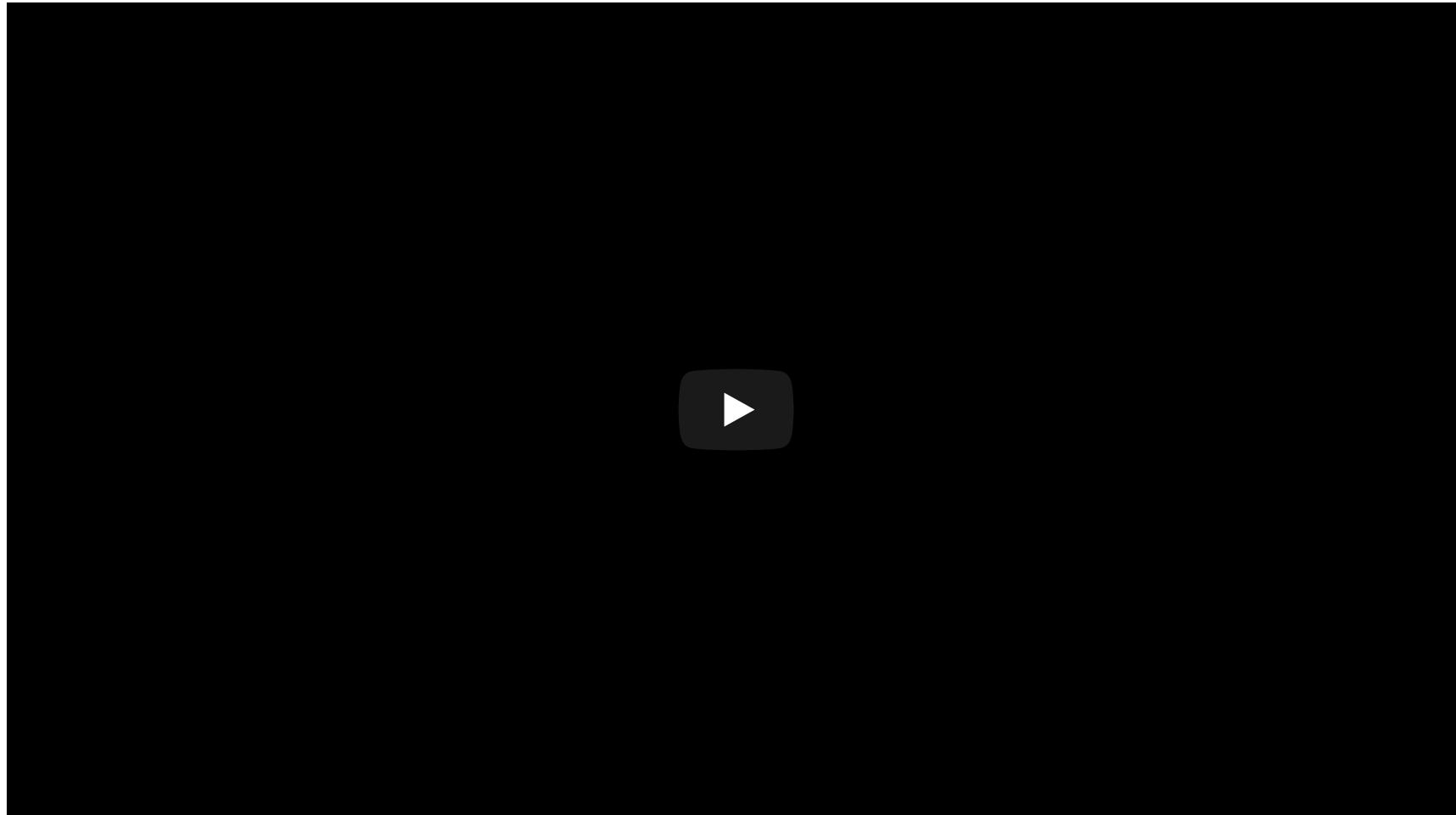
- But, GNU 프로젝트에는 커널이 없었고..

# Kernel



- 하드웨어와 응용프로그램을 이어주는 운영체제의 핵심 시스템소프트웨어

# Linus Torvalds



- 헬싱키 대학생이던 리누스 토발즈는 앤디 타넨바움의 MINIX를 개조한 Linux를 발표
- 0.1 - bash(GNU Bourne Again SHell), gcc(UNIX 기반 C 컴파일러)

# Linux

- 리누스 토발즈가 작성한 커널 혹은 GNU 프로젝트의 라이브러리와 도구가 포함된 운영체제
- PC와 모바일, 서버, 임베디드 시스템 등 다양한 분야에서 활용
- Redhat, Debian, Ubuntu, Android 등 다양한 배포판이 존재

# Shell

- 운영체제의 커널과 사용자를 이어주는 소프트웨어
- sh(Bourne Shell): AT&T Bell 연구소의 Steve Bourne이 작성한 유닉스 쉘
- csh: 버클리의 Bill Joy가 작성한 유닉스 쉘(C언어랑 비슷한 모양)
- bash(Bourne Again Shell): Brian Fox가 작성한 유닉스 쉘
  - 다양한 운영체제에서 기본 쉘로 채택
- zsh: Paul Falstad가 작성한 유닉스 쉘
  - sh 확장형 쉘
  - 현재까지 가장 완벽한 쉘

# Let's learn bash

# Shell Command Basic

```
$ cd documents  
  
$ mkdir python – make directory python  
$ cd python – change directory  
$ cd .. – up to  
  
$ ls  
$ ls -al  
  
$ touch hello.py – create hello.py  
$ exit – terminate shell
```

# chmod

| 파일의 권한을 설정할 때 사용

drwxr-xr-x

d or - : directory or file

(user)(group)(other)

r : read

w : write

x : execute

- : no permission

# chmod

```
$ chmod [옵션] (8진수) (파일명)
```

8진수

0: 000

1: 001

2: 010

3: 011

4: 100

5: 101

6: 110

7: 111

# Shell Command Basic

```
$ mv hello.py python  
$ cp hello.py python
```

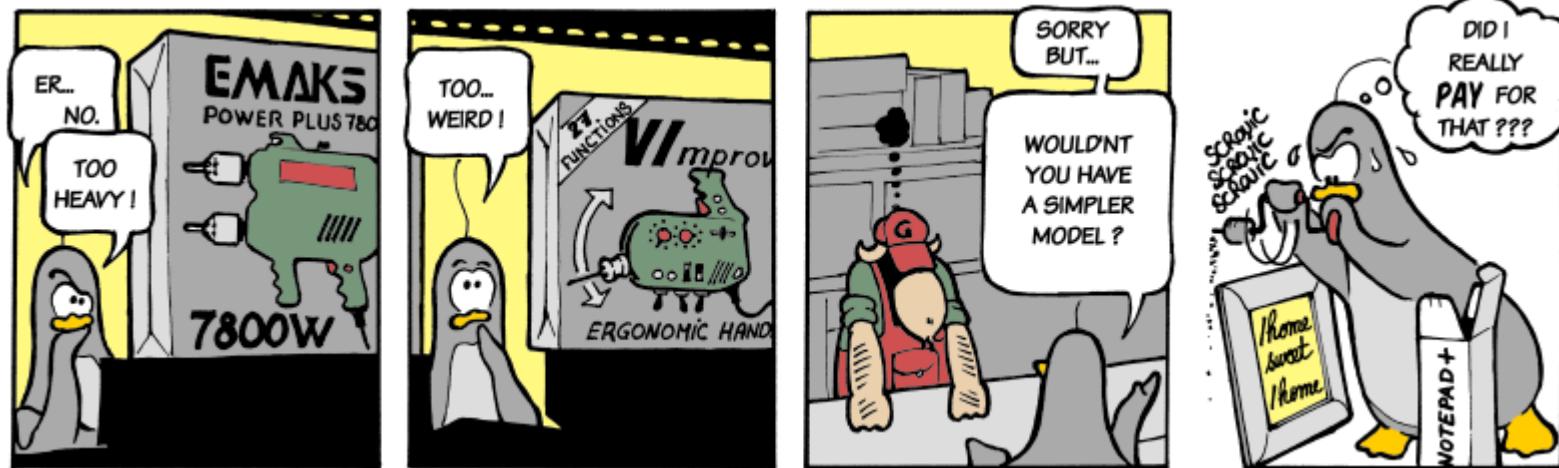
```
$ rm hello.py  
$ rm -rf python/
```

```
$ python --version  
$ python --help
```

# Vim



# Vim



Copyright (c) 2007 Laurent Gregoire

- Vi improved Text Editor

# Vim Basic

## Command

```
h,j,k,l – move cursor  
i – insert mode  
v – visual mode  
d – delete  
y – yank  
p – paste  
u – undo  
r – replace  
$ – move end of line  
^ – move start of line  
  
:q – quit  
:q! – quit w/o write(no warning)  
:wq – write and quit  
  
:{number} – move to {number}th line
```

write `hello.py` with Vim

```
$ vim
```

```
$ vim hello.py
```

```
i
```

```
-- insert --
```

type `print("hello python!")`

press `esc` to escape

```
:wq
```

```
$ python hello.py
```

## copy & paste

```
$ vim hello.py
```

```
v
```

```
-- visual --
```

```
블록지정 후 y
```

```
p
```

```
press esc to escape
```

```
:wq
```

```
$ python hello.py
```

## Use macro with Vim

```
$ vim hello.py
```

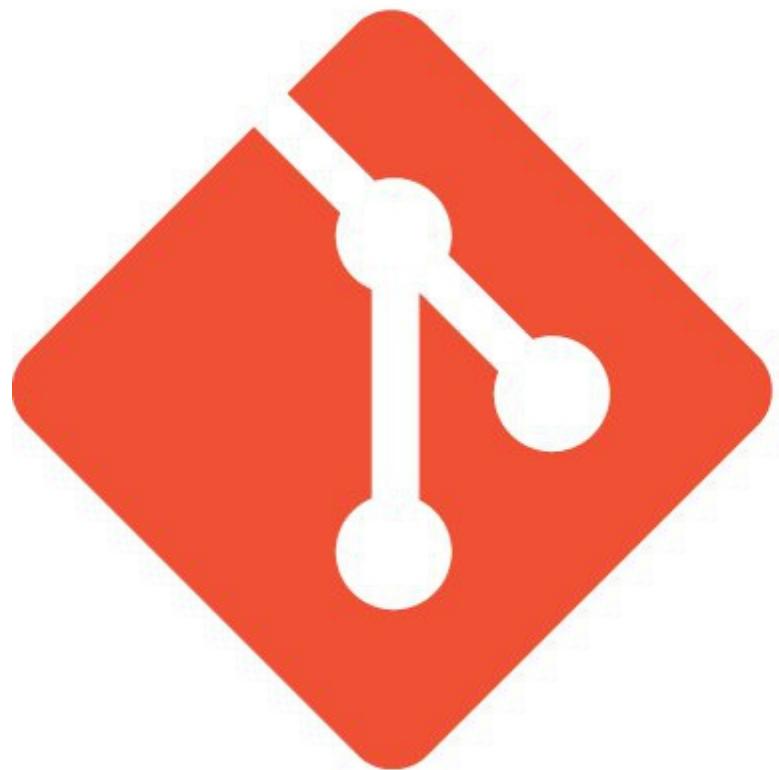
qa - a라는 매크로를 생성

--recording-- 이 보이면 매크로 작성

q - 매크로 작성 종료

@a - a 매크로 실행

10@a - a 매크로 10회 실행



git

# VCS (Version Control System)

== SCM (Source Code Management)

< SCM (Software Configuration Management: 형상관리)

# chronicle of git



## chronicle of git

- Linux Kernel을 만들기 위해 Subversion을 쓰다 화가 난 리누스 토발즈는 2주만에 git이라는 버전관리 시스템을 만듦  
[git official repo](#)

# Characteristics of git

- 빠른속도, 단순한 구조
- 분산형 저장소 지원
- 비선형적 개발(수천개의 브랜치) 가능

## Pros of git

- 중간-발표자료\_최종\_진짜최종\_15-4(교수님이 맘에들어함)\_언제까지??\_이걸로 갑시다.ppt
- 소스코드 주고받기 없이 동시작업이 가능해져 생산성이 증가
- 수정내용은 commit 단위로 관리, 배포 뿐 아니라 원하는 시점으로 Checkout 가능
- 새로운 기능 추가는 Branch로 개발하여 편안한 실험이 가능하며, 성공적으로 개발이 완료되면 Merge하여 반영
- 인터넷이 연결되지 않아도 개발할 수 있음

# Open-source project

<https://github.com/python/cpython>

<https://github.com/tensorflow/tensorflow>

<https://github.com/JuliaLang/julia>

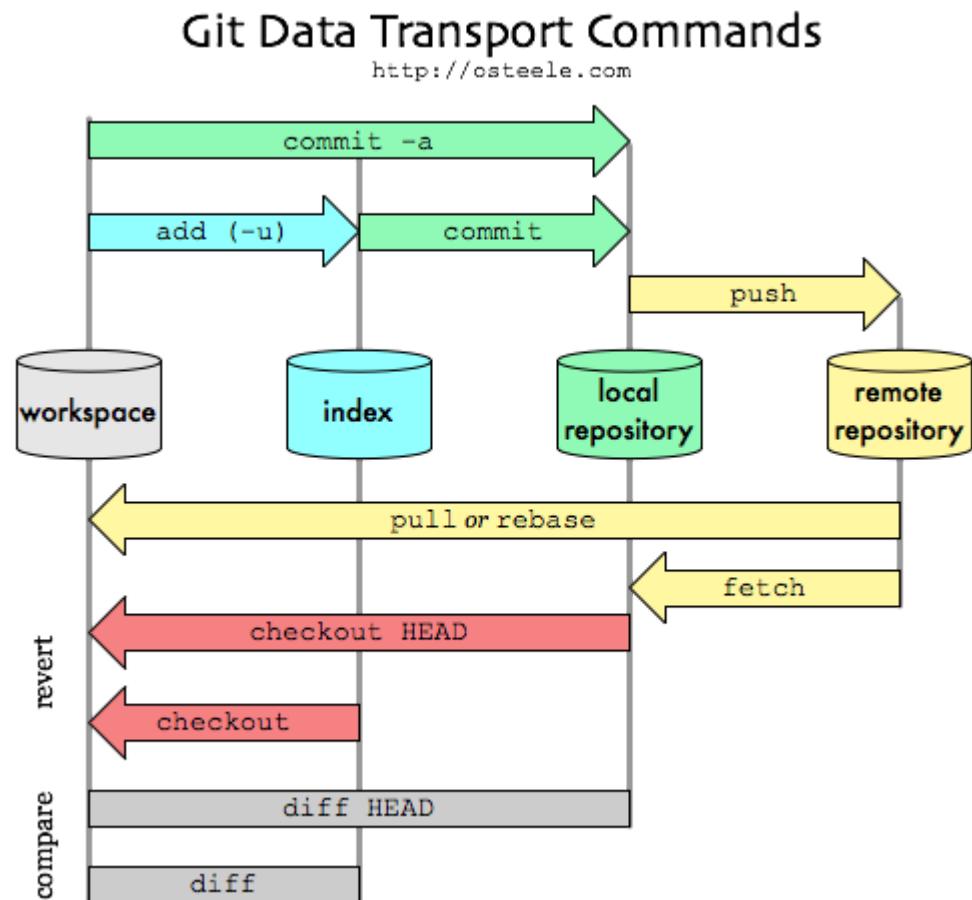
<https://github.com/golang/go>

<https://github.com/jkeun>

# git inside

- Blob: 모든 파일이 Blob이라는 단위로 구성
- Tree: Blob(tree)들을 모은 것
- Commit: 파일에 대한 정보들을 모은 것

# git Process and Command



## Useful manager for mac

[http://brew.sh/index\\_ko.html](http://brew.sh/index_ko.html)

## install git

<https://git-scm.com/>

```
// Mac OS  
$ brew install git  
// Linux  
$ sudo apt-get install git
```

- Windows: [install git bash](#)

`$ git --version` 으로 정상적으로 설치되었는지를 확인

# git is not equal to github



sign up github

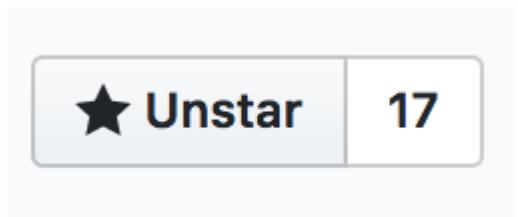
<https://github.com/>

important!!

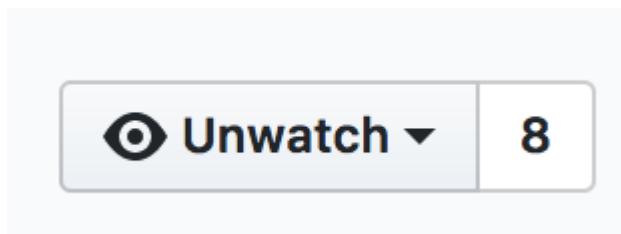
- 가입할 email 과 username 은 멋지게
- private repo를 원한다면 \$7/month

# Important github User Interface

# Star



# watch



# Set configuration

terminal

```
$ git config --global user.name "username"  
$ git config --global user.email "github email address"  
$ git config --list
```

# My First Repo

Let's make your first repo with github

# My First Repo

```
$ git init  
$ git add .  
$ git commit -m "some commit"
```

After create new repo through github,

```
$ git remote add origin https://github.com/username/repo.git  
$ git push origin master
```

# github pages

# My First Github Pages

github 저장소를 활용해 정적인 사이트 호스팅이 가능

username .github.io

<http://tech.kakao.com/>

<https://spoqa.github.io/>

## sample index page

After create new repo through github,

```
$ git clone https://github.com/username/username.github.io.git
```

Create New file `index.html`

```
$ git add .
```

```
$ git commit -m "first page"
```

```
$ git push origin master
```

## sample index page

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My first gh page</title>
  </head>
  <body>
    <h1>Home</h1>
    <p>Hello, there!</p>
  </body>
</html>
```

# Static Site Generator

- [Jekyll](#): Ruby 기반 정적인 블로그 생성기
  - 설치와 사용이 쉬움
  - 사용자가 많았음
- [Hugo](#): Golang 기반 정적인 블로그 생성기
  - 빠른 속도로 사이트를 생성
  - 사용자 증가 중
- [Hexo](#): Node.js 기반 정적인 블로그 생성기
  - Node.js를 안다면 커스터마이즈가 쉬움
  - 빠른 속도로 사용자 증가 중

## Recommand

Jekyll > Hugo > Hexo

# What is branch?

# What is branch?



MARVEL  
**GUARDIANS  
OF THE GALAXY**

fastcampus 프론트엔드 개발 스쿨, Wooyoung Choi, 2018

S SIDESHOW

# What is branch?

분기점을 생성하고 독립적으로 코드를 변경할 수 있도록 도와주는 모델

ex)

master branch

```
print('hello world!')
```

another branch

```
for i in range(1,10):
    print('hello world for the %s times!' % i)
```

# Branch

Show available local branch

```
$ git branch
```

Show available remote branch

```
$ git branch -r
```

Show available All branch

```
$ git branch -a
```

# Branch

Create branch

```
$ git branch stem
```

Checkout branch

```
$ git checkout stem
```

Create & Checkout branch

```
$ git checkout -b new-stem
```

make changes inside [readme.md](#)

```
$ git commit -a -m 'edit readme.md'
```

```
$ git checkout master
```

merge branch

```
$ git merge stem
```

# Branch

delete branch

```
$ git branch -D stem
```

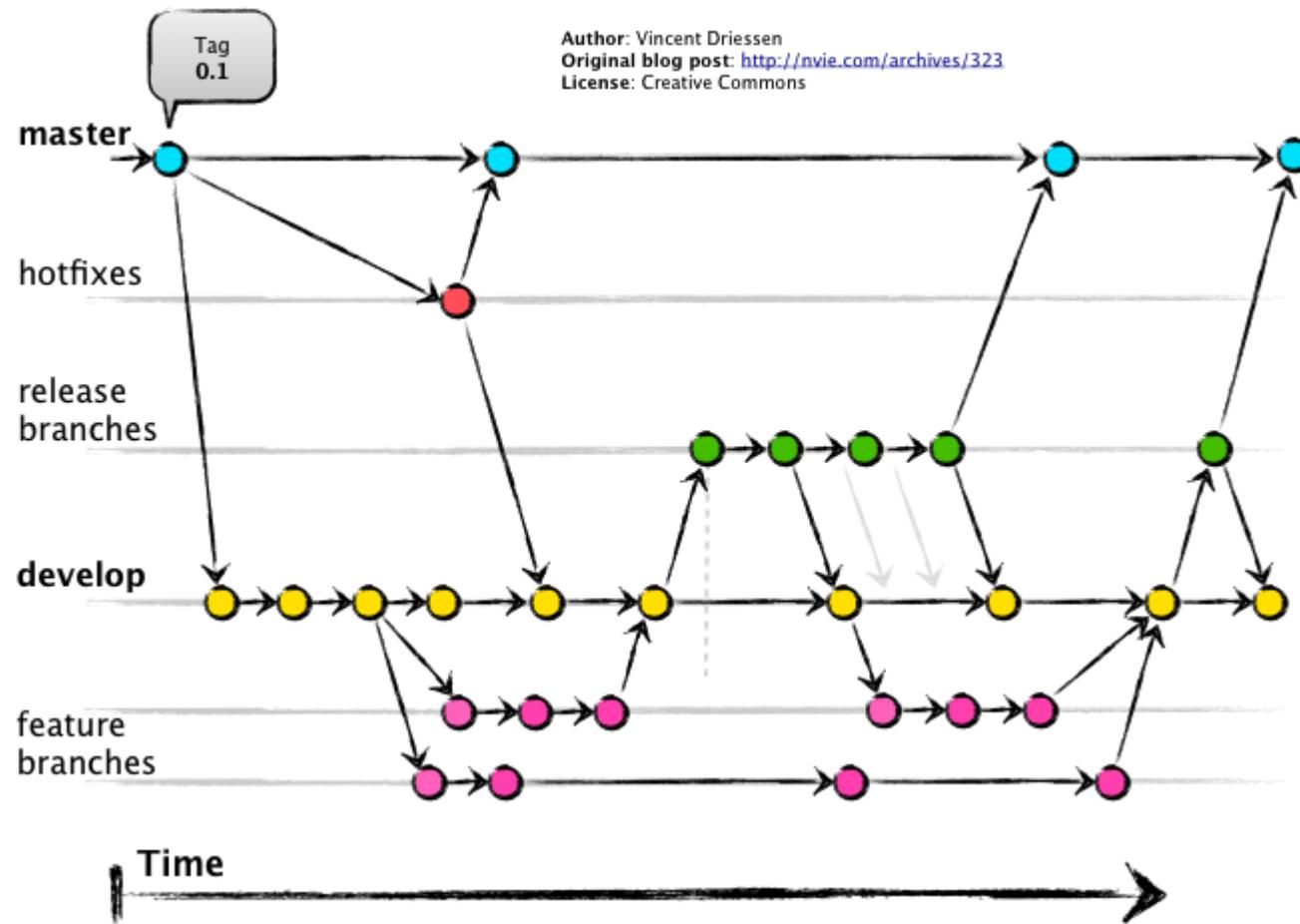
push with specified remote branch

```
$ git push origin stem
```

see the difference between two branches

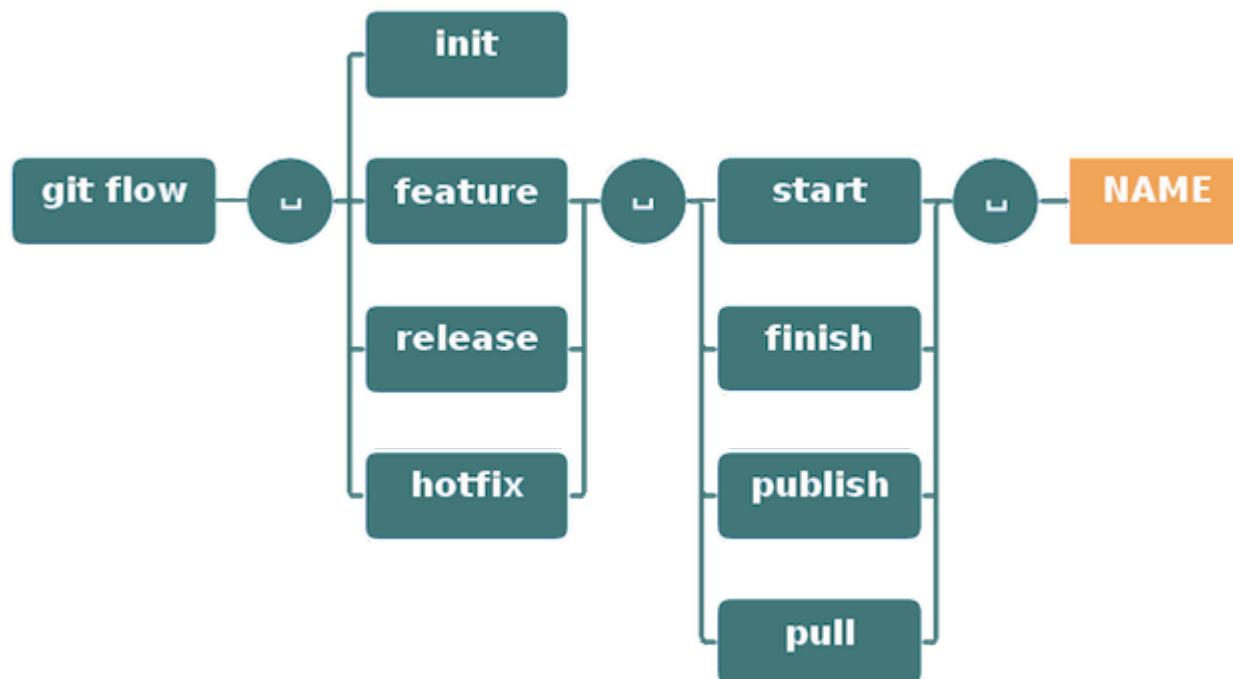
```
$ git diff master stem
```

# git flow strategy



# use git flow easily!

Link



# Collaborate with your Co-worker

# Method 1: Collaboration

## Add Collaborator

The screenshot shows the GitHub repository settings interface. The top navigation bar includes links for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Settings (which is highlighted with an orange border), and Insights. The left sidebar has sections for Options, Collaborators (which is also highlighted with an orange border), Webhooks, Integrations & services, and Deploy keys. The main content area is titled 'Collaborators' and contains the message: 'This repository doesn't have any collaborators yet. Use the form below to add a collaborator.' Below this is a search bar labeled 'Search by username, full name or email address' with the note: 'You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.' At the bottom right of the search bar is a button labeled 'Add collaborator'.

# Collaboration

Add, Commit and Push like you own it.

## Method 2: Fork and Merge

---

 Watch ▾ 1

 Unstar 1

 Fork 0

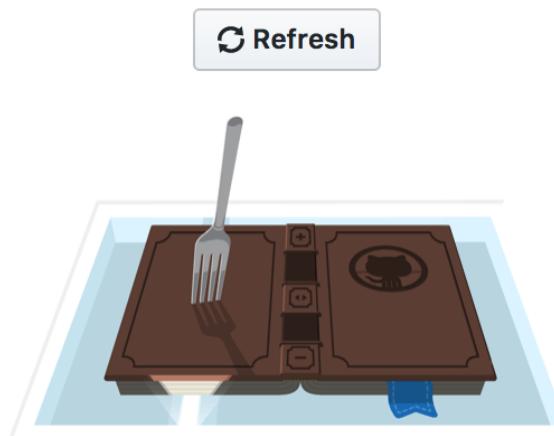
CS ▾

# Fork and Merge

---

## Forking JKeun/study-of-regression-toyota-corolla

It should only take a few seconds.



# Fork and Merge

 [ulgoon / study-of-regression-toyota-corolla](#)  
forked from [JKeun/study-of-regression-toyota-corolla](#)

 [Code](#)     [Pull requests 0](#)     [Projects 0](#)     [Wiki](#)    

 [Study - Regression Analysis using ToyotaCorolla dataset](#)  
[Add topics](#)

---

 [9 commits](#)     [1 branch](#)

---

 [Branch: master ▾](#)     [New pull request](#)

---

## Fork and Merge

```
$ git clone https://github.com/username/forked-repo.git  
$ git remote add upstream  
https://github.com/anotheruser/original-repo.git
```

## Fork and Merge

```
$ git fetch upstream  
$ git merge upstream/master  
$ git branch -a  
$ git checkout -b new-feature
```

# Fork and Merge

Make some change

```
$ git add file  
$ git commit -m "commit message"  
$ git push origin new-feature
```

# Fork and Merge

No description, website, or topics provided.

Edit

Add topics

1 commit

3 branches

0 releases

1 contributor

Your recently pushed branches:

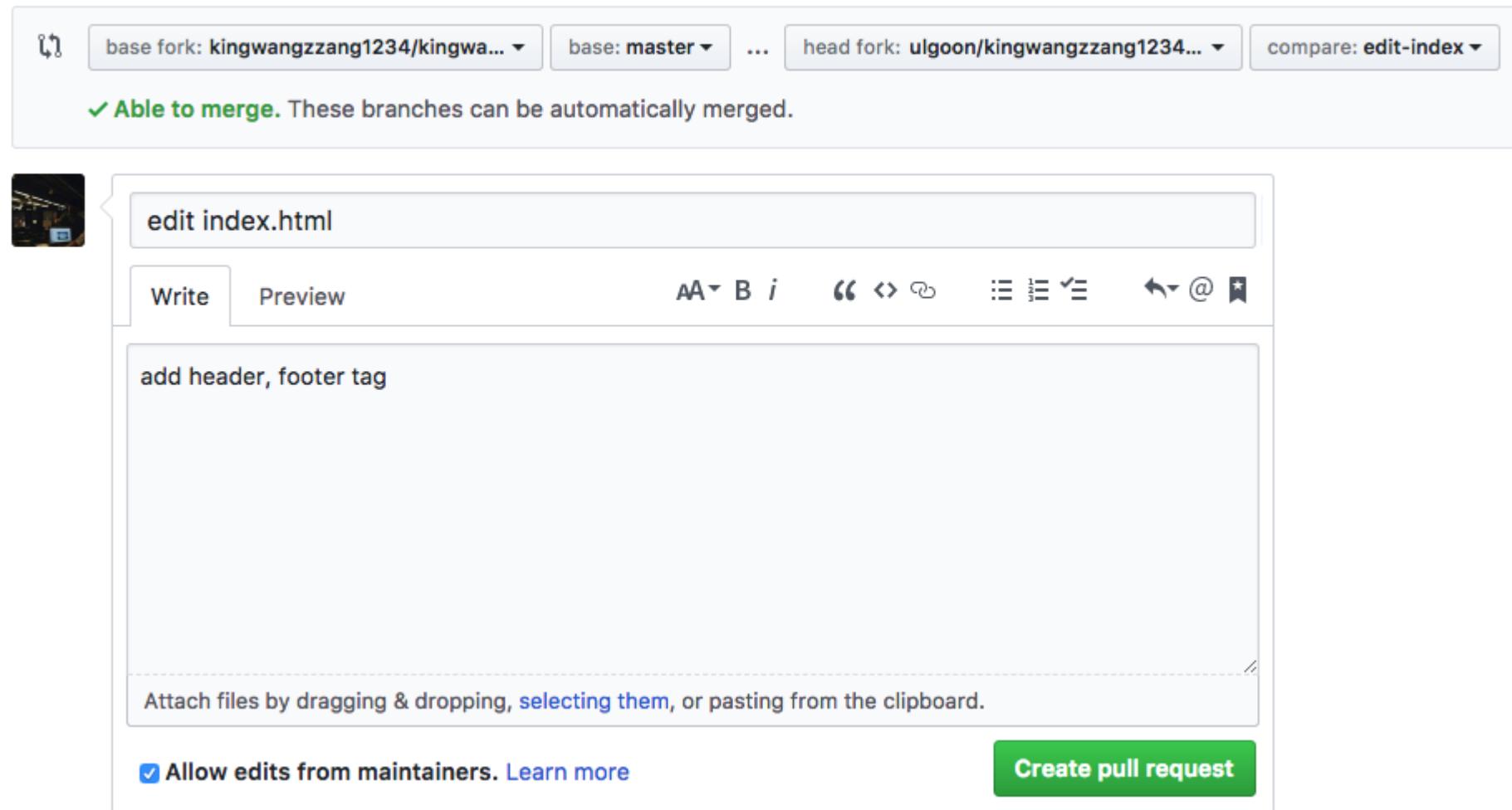
edit-index (less than a minute ago)

Compare & pull request

# Fork and Merge

## 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](#).



The screenshot shows the GitHub interface for opening a pull request. At the top, there are dropdown menus for 'base fork' (set to 'kingwangzzang1234/kingwa...'), 'base' (set to 'master'), '...', 'head fork' (set to 'ulgoon/kingwangzzang1234...'), and 'compare' (set to 'edit-index'). A green checkmark indicates that the branches are 'Able to merge'. Below this, the 'edit index.html' file is shown with the content 'add header, footer tag'. There are tabs for 'Write' and 'Preview', and a toolbar with various icons. A note at the bottom says 'Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.' At the bottom left is a checkbox for 'Allow edits from maintainers' with a link to 'Learn more'. On the right is a large green button labeled 'Create pull request'.

# Fork and Merge

## edit index.html #2

 Open ulgoon wants to merge 1 commit into `kingwangzzang1234:master` from `ulgoon:edit-index`

Conversation 0 Commits 1 Files changed 1

 ulgoon commented 17 seconds ago

Contributor + 

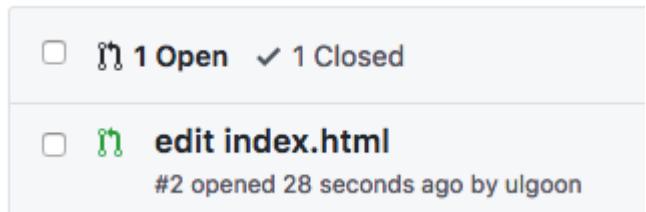
add header, footer tag

 edit index.html ... d81b362

Add more commits by pushing to the **edit-index** branch on [ulgoon/kingwangzzang1234.github.io](#).

  This branch has no conflicts with the base branch  
Only those with [write access](#) to this repository can merge pull requests.

# Fork and Merge



# Fork and Merge

## edit index.html #2

**Open** ulgoon wants to merge 1 commit into `kingwangzzang1234:master` from `ulgoon:edit-index`

Conversation 0 Commits 1 Files changed 1

ulgoon commented 38 seconds ago

Contributor +

add header, footer tag

edit index.html ... d81b362

Add more commits by pushing to the `edit-index` branch on [ulgoon/kingwangzzang1234.github.io](#).

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](#).

# Fork and Merge

## edit index.html #2

**Open** ulgoon wants to merge 1 commit into `kingwangzzang1234:master` from `ulgoon:edit-index`

Conversation 0    Commits 1    Files changed 1

ulgoon commented 38 seconds ago

Contributor + 🧑‍💻 🖊

add header, footer tag

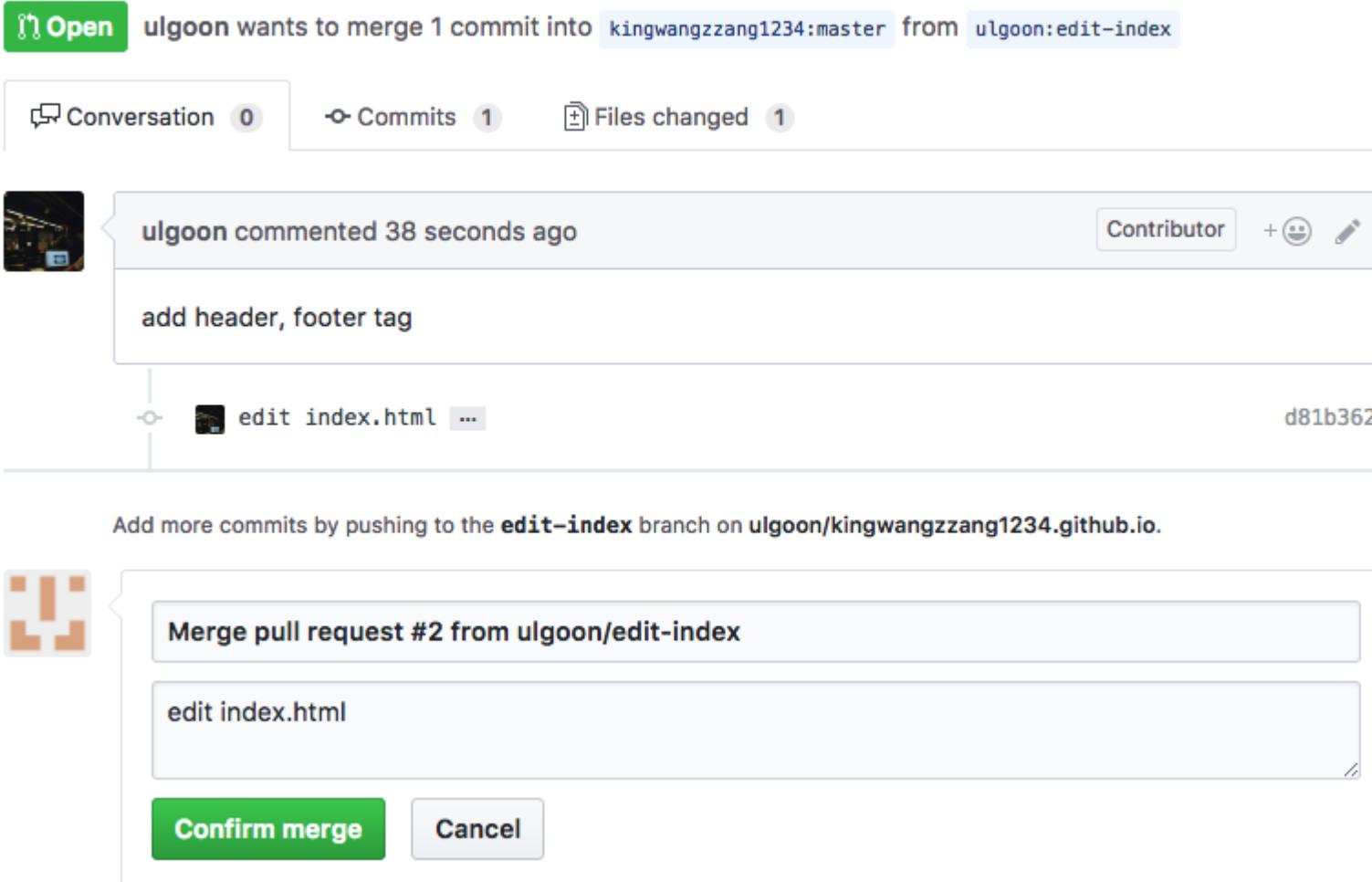
edit index.html ... d81b362

Add more commits by pushing to the `edit-index` branch on `ulgoon/kingwangzzang1234.github.io`.

Merge pull request #2 from `ulgoon/edit-index`

edit index.html

Confirm merge Cancel

A screenshot of a GitHub pull request interface. At the top, there's a green 'Open' button and a message from 'ulgoon' about merging a commit from their 'edit-index' branch into the 'master' branch of 'kingwangzzang1234'. Below this are tabs for 'Conversation' (0), 'Commits' (1), and 'Files changed' (1). A comment from 'ulgoon' is shown, mentioning adding a header and footer tag. Below the comment is a commit for 'edit index.html' with hash 'd81b362'. A note at the bottom says to add more commits by pushing to the 'edit-index' branch on 'ulgoon/kingwangzzang1234.github.io'. A modal dialog box is open, prompting to 'Merge pull request #2 from ulgoon/edit-index' and listing the file 'edit index.html'. It has 'Confirm merge' and 'Cancel' buttons.

# Fork and Merge

## edit index.html #2

Merged kingwangzzang... merged 1 commit into kingwangzzang1234:master from ulgoon:edit-index just now

Conversation 0 Commits 1 Files changed 1

ulgoon commented 38 seconds ago

add header, footer tag

Contributor + 😊 🖊

-o edit index.html ... d81b362

kingwangzzang1234 merged commit 45d71fa into kingwangzzang1234:master just now

Revert

A screenshot of a GitHub pull request page. The title is "edit index.html #2". A purple button at the top left says "Merged". Below it, a message from "ulgoon" says "commented 38 seconds ago" with the message "add header, footer tag". There's a "Contributor" button with a plus sign, a smiley face emoji, and a pencil icon. Below the comment, there's a commit log entry for "edit index.html" with hash "d81b362". At the bottom, another commit is shown where "kingwangzzang1234" merged "commit 45d71fa" into "kingwangzzang1234:master" "just now". A "Revert" button is next to this merge commit.

## continuous pull

## continuous pull

```
$ git remote add upstream
```

```
https://github.com/anotheruser/original-repo.git
```

```
$ git fetch upstream
```

```
$ git merge upstream/master
```

# Assignment

[Try git](#)

마지막 결과를 Print Screen 후 조교님께 제출해주세요.