

Fastcampus Programming SCHOOL

Shell, vim Command && git

Introduce

최우영

- Co-founder, CTO(disceptio)
- Solution Architect, Web Developer, Instructor
- Skills: Python, Golang, Julia, Node.js, Google tag manager ...

blog: <https://ulgoon.github.io/>

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

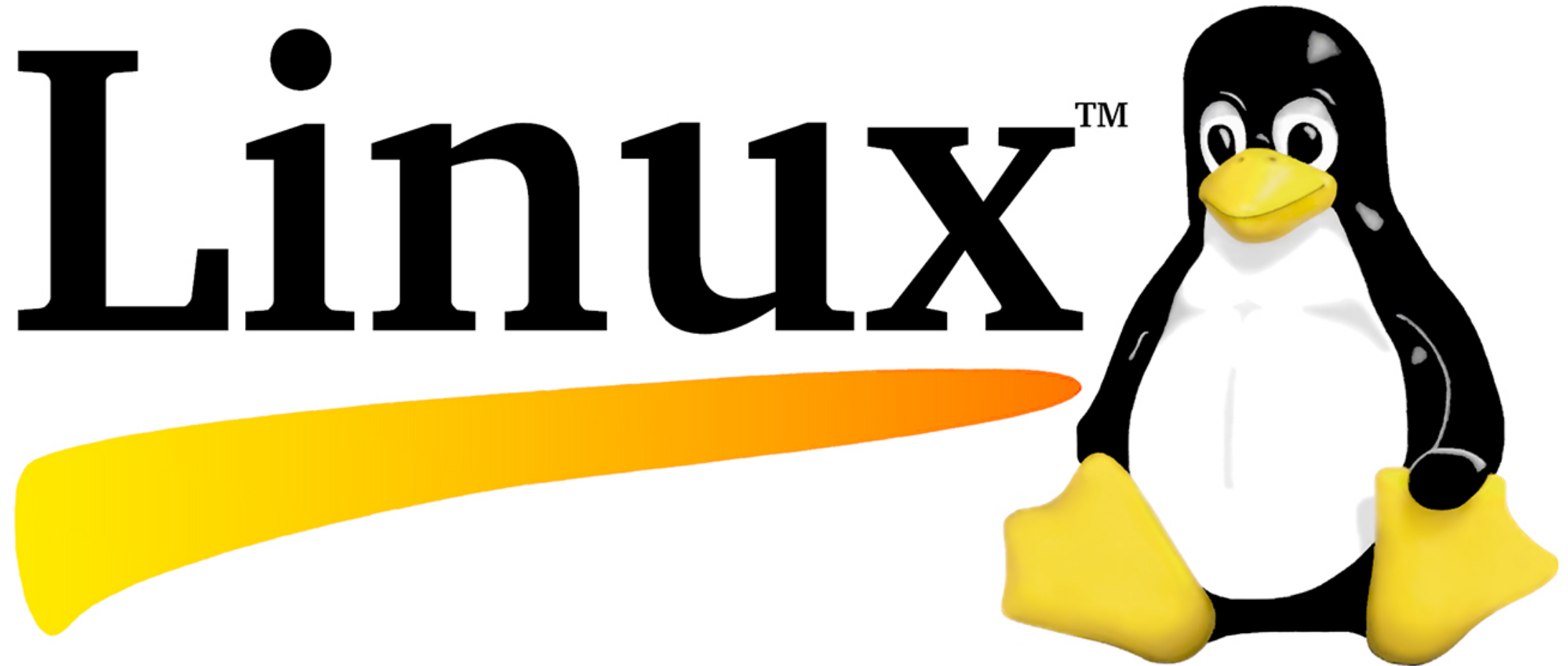
email: me@ulgoon.com

Goal

- Linux의 역사를 이해한다
- CLI에 대한 공포를 극복하고 Shell과 친구가 된다
- Linux Shell 커맨드를 학습하여 능숙하게 이를 활용할 수 있다
- Vim 텍스트 에디터를 통해 파일을 작성하고 매크로를 만들 수 있다

Goal

- git을 이해하고, git과 github이 다름을 인지한다
- git을 활용하여 나의 소스코드를 관리할 수 있다
- 나의 커리어를 Swag 할 수 있는 블로그를 git을 활용하여 관리할 수 있다.
- git의 branch model을 활용해 능숙하게 코드관리할 수 있다
- git으로 타인과 협업하며, 다른 프로젝트에 기여할 수 있다



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

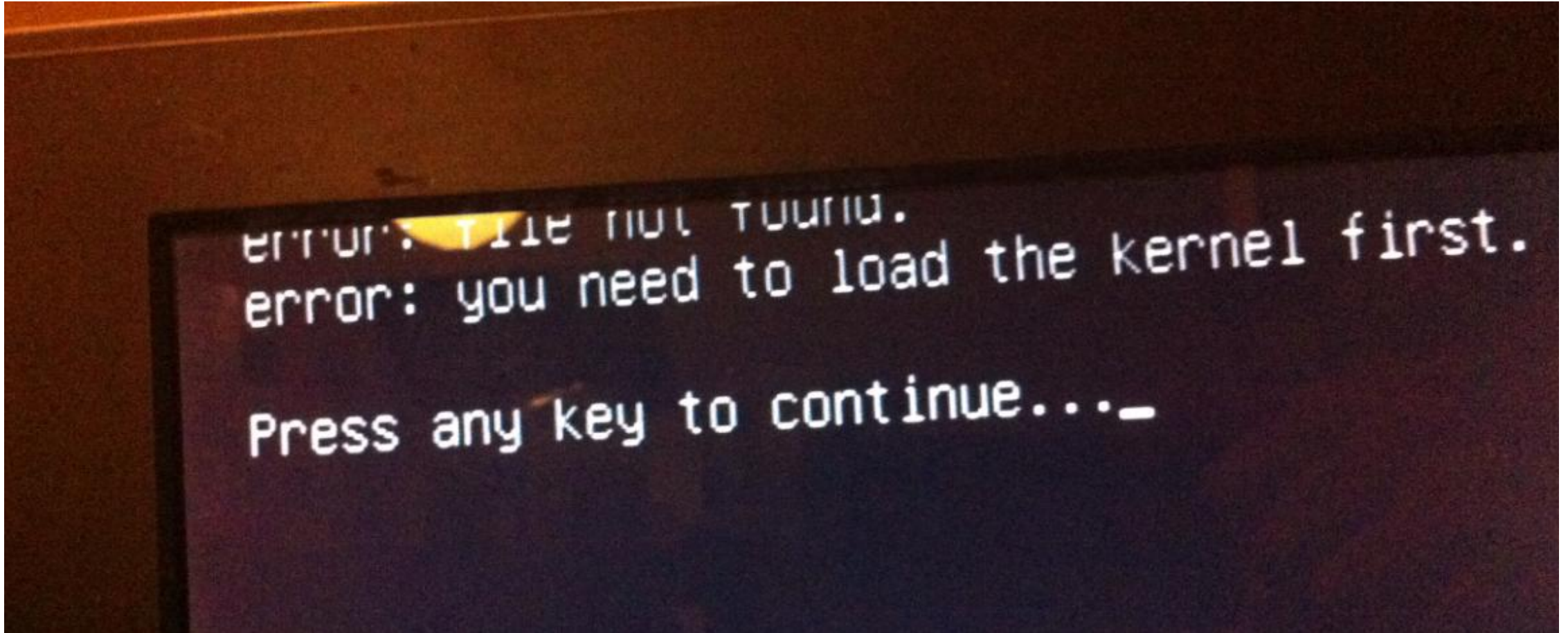


- 1984년 리처드 스톨먼이 오픈 소프트웨어 자유성 확보를 위한 GNU 프로젝트 돌입

Meaning of GNU

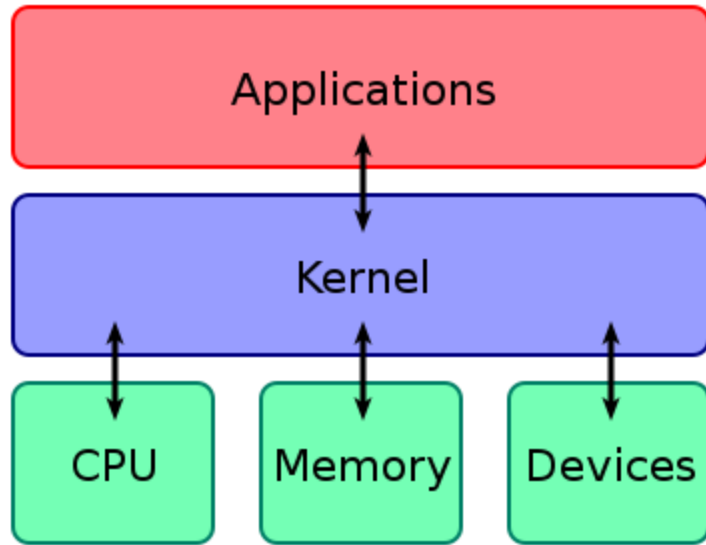
GNU == G NU is N ot U nix

Before Linux



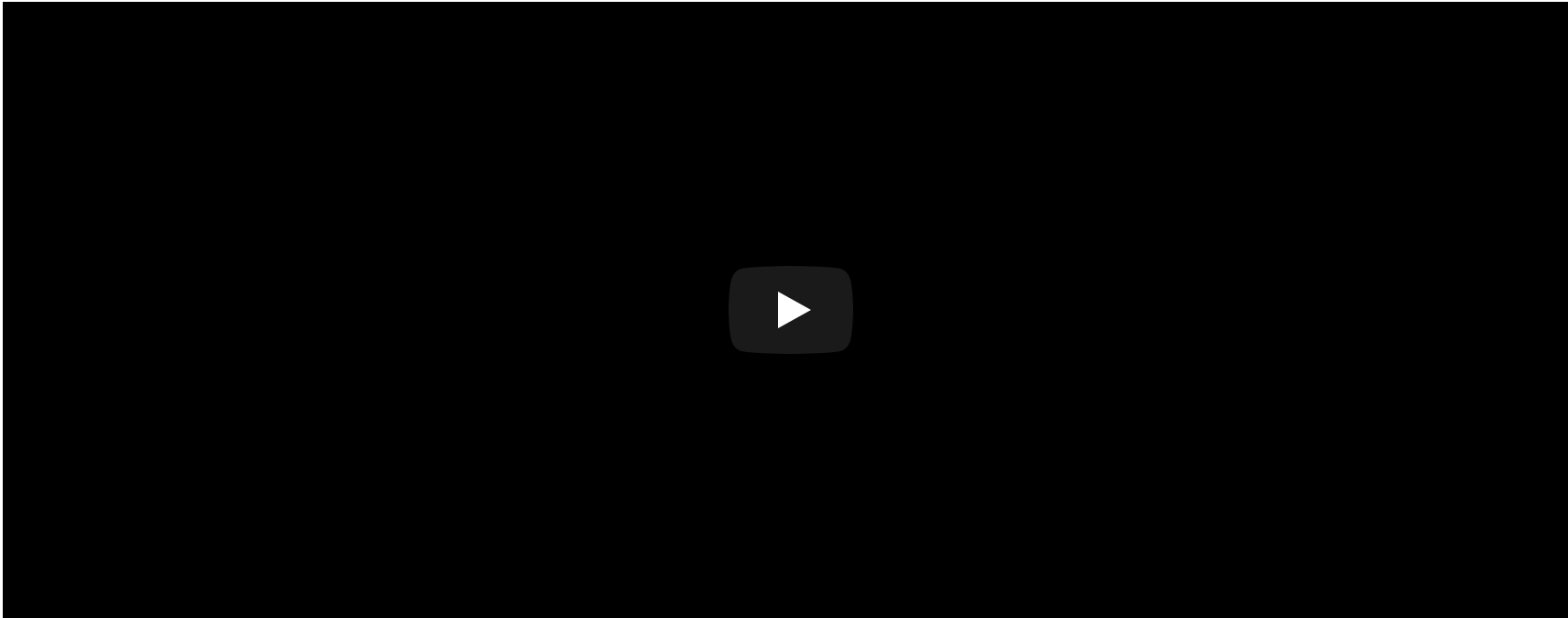
- But, GNU 프로젝트에는 커널이 없었고..

Kernel



- 하드웨어와 응용프로그램을 이어주는 운영체제의 핵심 시스템소프트웨어

Linus Torvalds



<https://www.youtube.com/embed/IVpOyKCNZYw>

- 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 운영 체제
- 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 셸
- csh: 버클리의 Bill Joy가 작성한 유닉스 셸(C언어랑 비슷한 모양)
- bash(Bourne Again Shell): Brian Fox가 작성한 유닉스 셸
 - 다양한 운영체제에서 기본 셸로 채택
- zsh: Paul Falstad가 작성한 유닉스 셸
 - sh 확장형 셸
 - 현재까지 가장 완벽한 셸

Let's learn bash

Shell Command Basic

```
$ cd documents
```

```
$ mkdir dev # - make directory dev
```

```
$ cd dev # - change directory
```

```
$ cd .. # - go up
```

```
$ 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 dev
```

```
$ cp hello.py dev
```

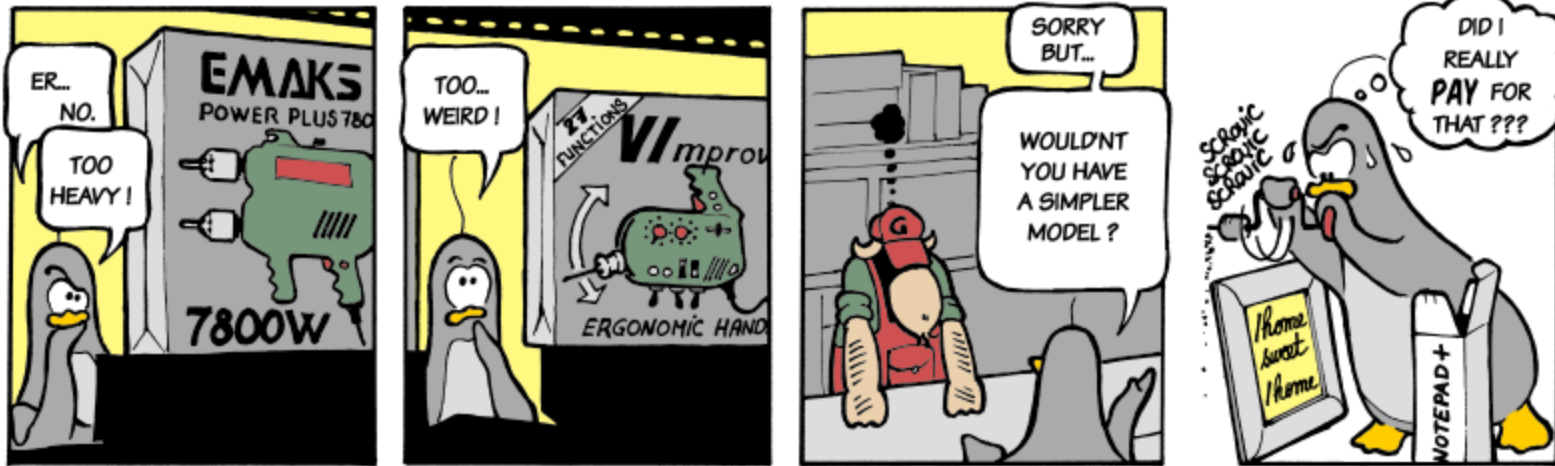
```
$ rm hello.py
```

```
$ rm -rf dev/
```

Vim



Vim



Copyright (c) 2007 Laurent Gregoire

- Vi improved Text Editor

TODO

1. documents/dev에서 fastcampus 디렉토리를 만듭니다
2. 새파일 index.html을 만들어주세요
3. vim으로 내부 작업을 한뒤 저장하여 cat 명령어로 저장을 확인합니다.
4. index.html을 style.css로 이름을 변경합니다.

Vim Basic

`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 Kernal을 만들기 위해 Subversion을 쓰다 화가 난 리누스 토발즈는 2주만에 git 이라는 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??_이걸로갑시다.ppt
- 소스코드 주고받기 없이 동시작업이 가능해져 생산성이 증가
- 수정내용은 **commit** 단위로 관리, 배포 뿐 아니라 원하는 시점으로 **Checkout** 가능
- 새로운 기능 추가는 **Branch**로 개발하여 편한 실험이 가능하며, 성공적으로 개발이 완료되면 **Merge**하여 반영
- 인터넷이 연결되지 않아도 개발할 수 있음

git GUI Clients

- git GUI
- sourcetree
- kraken
- smartGit

CLI first!

- Source code를 Cloud Platform에서 수행할 경우, CLI 커맨드로 버전관리를 수행해야 합니다.
- CLI 커맨드로 git을 사용할 줄 알면, GUI 도구가 제공하는 기능에 대한 이해가 빠릅니다.
- 확인용도로 GUI를 참고하는 것은 Good^^

Open-source project

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

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

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

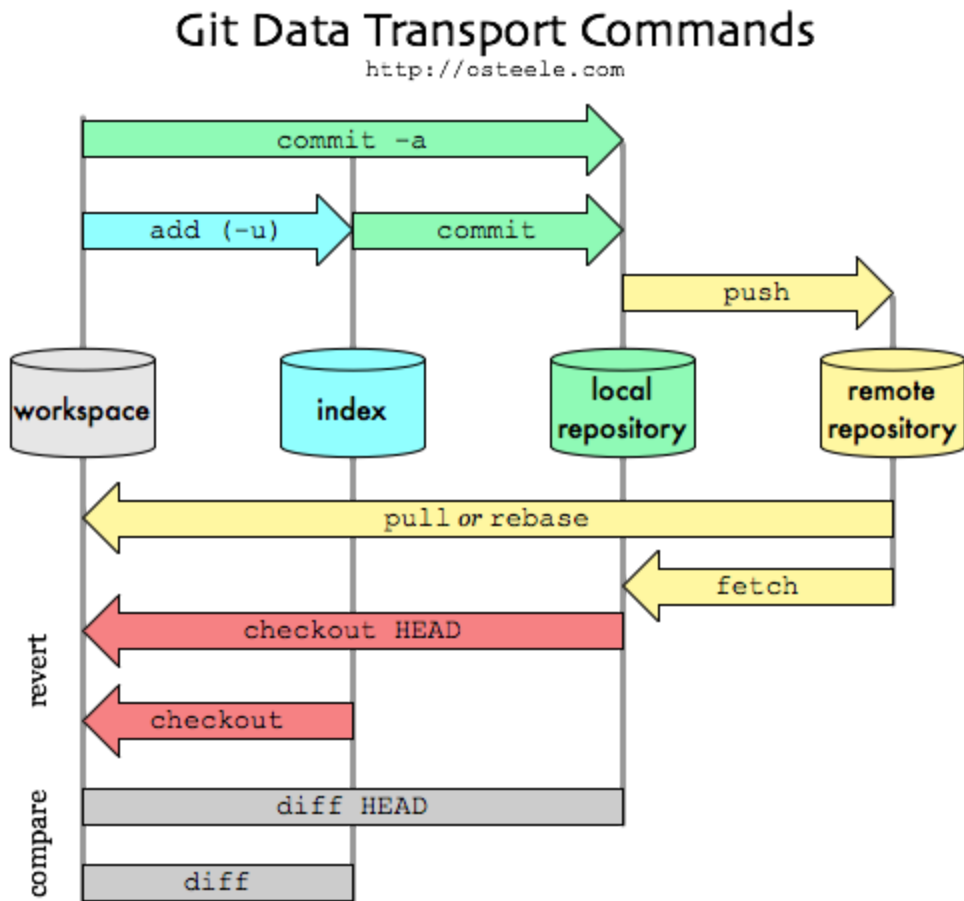
<https://github.com/golang/go>

<https://github.com/facebook/react>

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

install git

- for windows

<https://gitforwindows.org/>

// MacOS

```
$ brew install git
```

// Linux

```
$ sudo apt-get install git
```

```
$ git --version
```

 으로 정상적으로 설치되었는지를 확인

git is not equal to github



sign up github

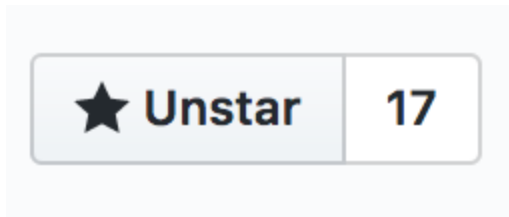
<https://github.com/>

important!!

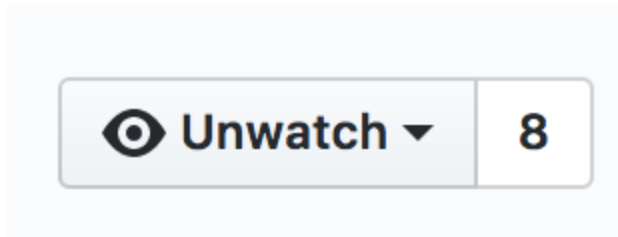
- 가입할 email 과 username 은 멋지게
- private repo를 원한다면 \$7/month

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 --global core.editor "vim"  
$ git config --list
```

My First Repo

Let's make your first repo with github

My First Repo

After create new repo through github,

```
$ git init
```

```
$ git remote add origin https://github.com/{username}/{reponame}.git
```

```
$ git add .
```

```
$ git commit -m "some commit"
```

```
$ git push -u origin master
```

Second push to My First Repo

```
$ git add .  
$ git commit  
$ git push
```

start project with clone

- github에서 repo를 생성합니다.

```
$ git clone {repo address}  
$ git add .  
$ git commit  
$ git push
```


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 커스터마이징이 쉬움
 - 빠른 속도로 사용자 증가 중

Recommand

Hexo > Jekyll > Hugo

Let's use Hexo

Requirements

1. git
2. node.js(<https://nodejs.org/en/>)

```
$ npm install -g hexo-cli
```

Init hexo project

```
$ hexo init <folder>  
$ cd <folder>  
$ npm install
```

clean && generate static files

```
$ hexo clean && hexo generate
```

Run hexo server

```
$ hexo server
```

deploy

```
$ npm install hexo-deployer-git --save
```

```
deploy:  
  type: git  
  repo: <repository url>  branch: [branch] #published  
  message:
```

.gitignore and .gitattributes

.gitignore: 특정파일 추적을 하고 싶지 않을 경우

```
*.java  
*.py[cod]
```

.gitattributes: 파일단위, 디렉토리 별 다른 설정을 부여하고 싶을 경우

```
# Avoid conflicts in pbxproj files  
.pbxproj merge=union  
  
# Always diff strings files as text  
*.strings text diff
```


What is branch?

What is branch?



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
```

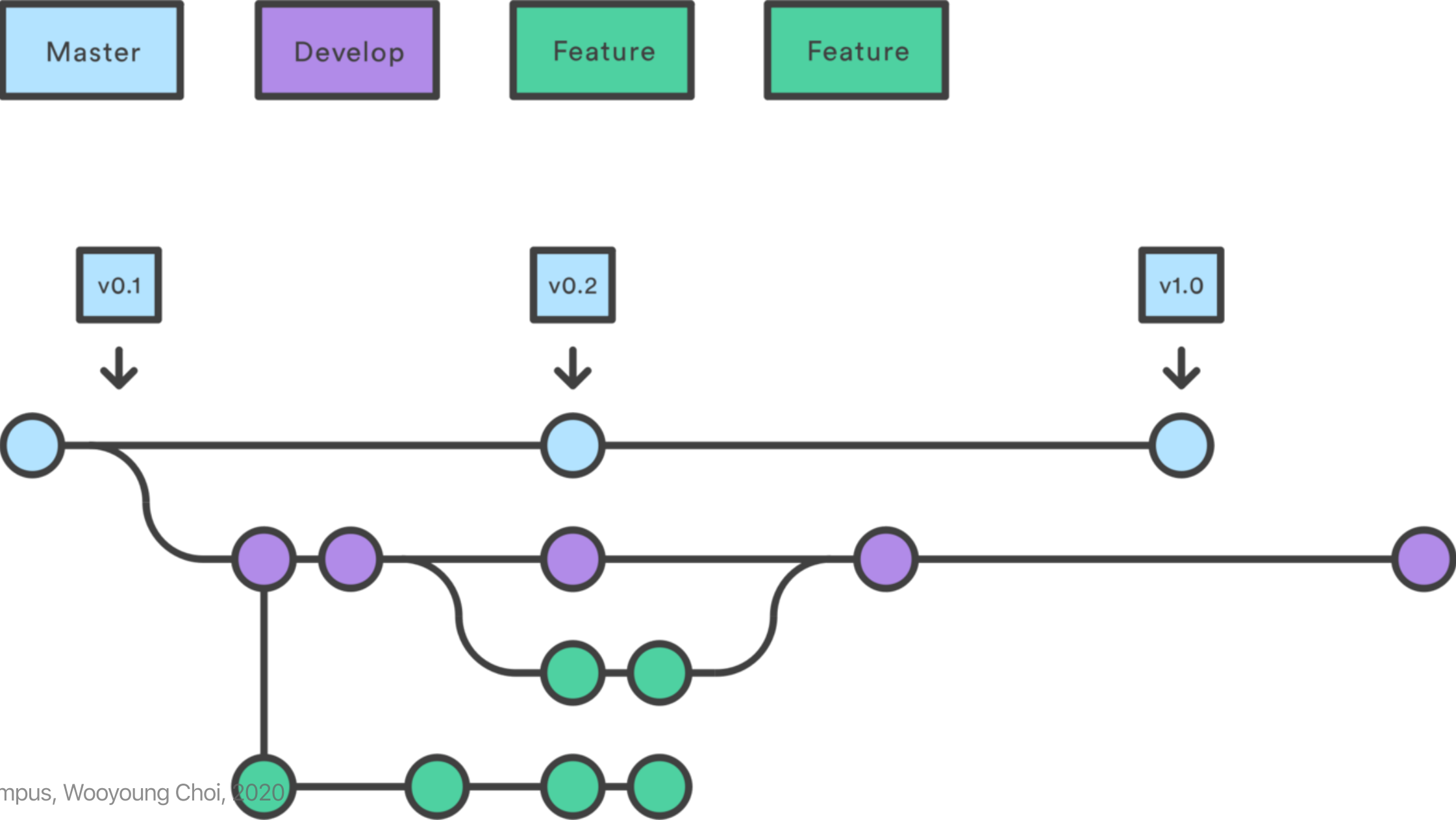
push with specified remote branch

```
$ git push origin stem
```

see the difference between two branches

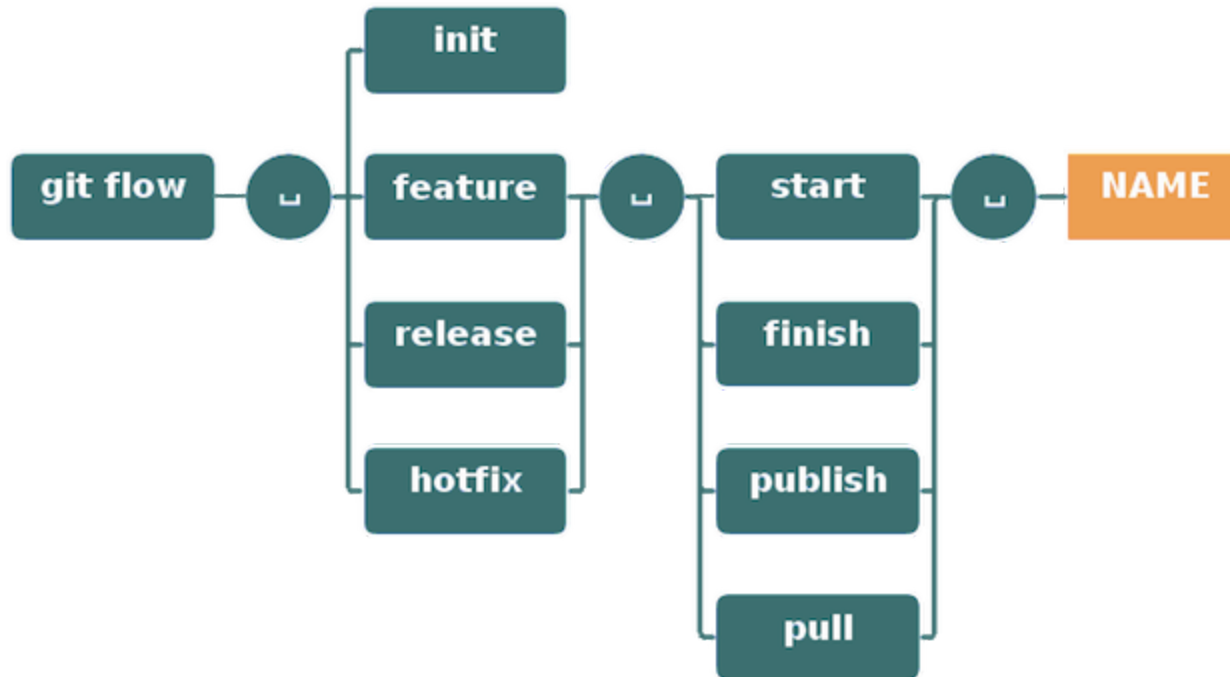
```
$ git diff master stem
```

git flow strategy



use git flow easily!

[Link](#)



Collaborate with your Co-worker

Method 1: Collaboration

Add Collaborator

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Settings](#) [Insights ▾](#)

[Options](#)
Collaborators
[Webhooks](#)
[Integrations & services](#)
[Deploy keys](#)

Collaborators

Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

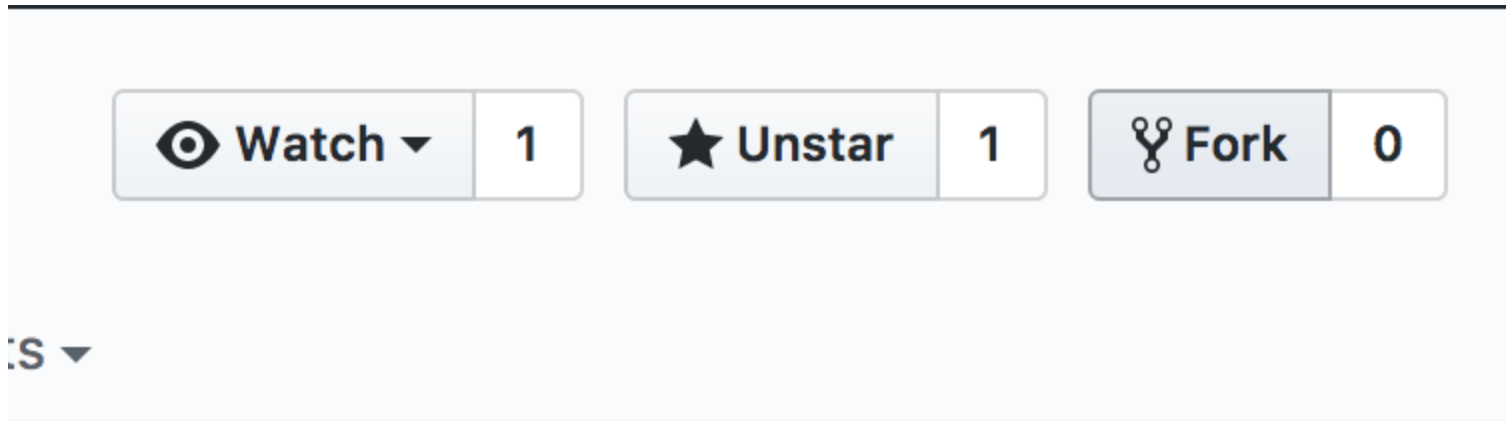
Search by username, full name or email address
You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

Add collaborator

Collaboration

Add, Commit and Push like you own it.

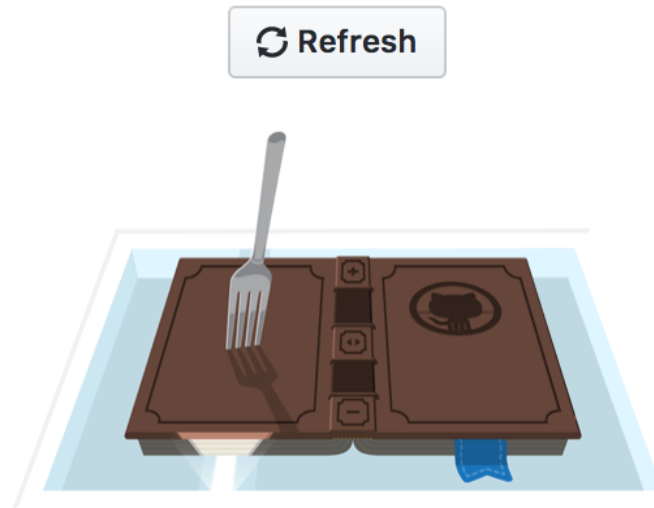
Method 2: Fork and Merge



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
```

Fork and Merge

```
$ 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](#).



base fork: kingwangzzang1234/kingwa... ▾


base: master ▾

...

head fork: ulgoon/kingwangzzang1234... ▾

compare: edit-index ▾

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



edit index.html

Write

Preview

AA ▾ B i “ <> 🔗 ⋮ ≡ ✓≡ ↶ @ 🚩

add header, footer tag


Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.




☒ **Allow edits from maintainers.** [Learn more](#)

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](https://github.com/ulgoon/kingwangzzang1234).




This branch has no conflicts with the base branch

Only those with [write access](#) to this repository can merge pull requests.


Fork and Merge




☐  1 Open ✓ 1 Closed

☐  **edit index.html**
#2 opened 28 seconds ago by ulgoon

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](https://github.com/ulgoon/kingwangzzang1234).



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

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

Contributor



add header, footer tag



edit index.html ...

d81b362



kingwangzzang1234 merged commit 45d71fa into kingwangzzang1234:master just now

Revert

continuous pull

continuous pull

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

```
$ git fetch upstream
```

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