

Fastcampus

Frontend Dev SCHOOL

Software Engineering(3), gulp

개발자가 오프라인에서 살아가는 법

Communicate with Co-worker

상대를 아프게 하지 마세요

작은 것을 칭찬하세요

회사는 개발자만 존재하지 않아요

개발자의 의사소통 능력 - 지디넷코리아

www.zdnet.co.kr/column/column_view.asp?article_id... ▾ Translate this page

Jul 18, 2016 - 개발자의 의사소통 능력은 코딩실력보다 중요하다. 이미 여러 번 했던 이야기다.'개발자의 생명은 커뮤니케이션'이라는 칼럼에서 개발자의 의사소통이 정확히 무엇을 의미하는지도 설명했다. 이번 글은 그 내용의 확장판이다. 개발자가 좋은 의사소통을 하기 위해서 기억해야 하는 내용을 설명한다. . 1. 어머니에게 ...

개발자와 대화하는 법을 배워야 하는 이유 - beSUCCESS

<https://besuccess.com/.../why-do-we-learn-how-to-communicate-wit...> ▾ Translate this page

그 중에서도 관리자, 기획자가 일선에 서있는 소프트웨어 업계에서는 정말 심각한 문제이다. 문제를 심화시키는 요인은 암묵적이지만 통상적으로 인정하는 진리인, 개발자들이 사회생활, 의사소통에 그리 능숙하지 않다는 점이다. 하지만 그들이 논리적으로 자신의 불만을 표현하지 못한다고 해서 그것을 무시해도 된다는 것은 아니다.

개발자가 비개발자와의 의사소통을 배워야 하는 이유. :: eedler

blog.eedler.com/22 ▾ Translate this page

Oct 7, 2016 - 개발자가 비개발자와의 의사소통을 배워야 하는 이유. 예전에 일했던 회사에서 인사담당자가 이런저런 사람을 뽑으면서 했던 말이 생각납니다. 다른 기업들이 왜 말도 안되고, 확인도 잘 안되는 조건인 '의사소통..

개발자가 익혀야 할 업무스킬 | IT의 중심에서

<https://subokim.wordpress.com/2011/12/01/job-skill-of-developers/> ▾ Translate this page

Dec 1, 2011 - 하지만 회사라는 장소에 오면, 디자이너와의 회의, 상사 보고 등 "업무 스킬(Business Job Skill)"도 매우 중요하게 됩니다. 일반적으로 많은 개발자들이 How(어떻게 해야 구현이 될 것인가?)에 대해서 정통하지만, What(최종적으로 무엇을 만들고 있나?)이나 Why(왜 이것을 만들고 있나?)에 대해서는 무관심합니다.

개발자스럽다 - 의사소통에 관해 유독 개발자 이야기가 많이 나온다. 많이 ...

https://www.facebook.com/gaeraecom/posts/1005159469605475?comment_tracking...

의사소통에 관해 유독 개발자 이야기가 많이 나온다. 많이 나오고 있다면 그만한 이유도 있을 거다. 물론 상대방도 개발자를 이해하려는 노력도 있었으면 좋겠다. 복잡한 로직을 쉽게 설명하는 능력은 쉬운 능력이 아니다. #개발자 #소통 #커뮤니케이션 #communication...

르세상스 엔지니어 & 창업가 :: 의사소통 능력이 부족한 개발자

minslove.tistory.com/113 ▾ Translate this page

Aug 27, 2012 - 기술적인 열정을 가지고 있으며 기술력이 좋은 개발자는 이상하게 의사소통 능력이 떨어지는 경우가 많습니다. 따라서 회사에 필요한 가치를 많이 만들고 있으면서도 그런 것을 잘 드러내지 못합니다. 반면 의사소통에 보다 신경을 쓰는 개발자는 상대적으로 덜한 가치를 만들면서도 그것을 매우 잘 드러내고 그에 따른 ...

How를 생각하기 전 What과 Why를 먼저 생각하세요

404 보단 404 – Not Found

안되요 보단 시스템에 악영향을 줄 가능성이 있어 다른 방법을 고려하는 것이 좋을 것 같습니다.

개발자가 관리해야 할 것

- github
- linkedin
- blog

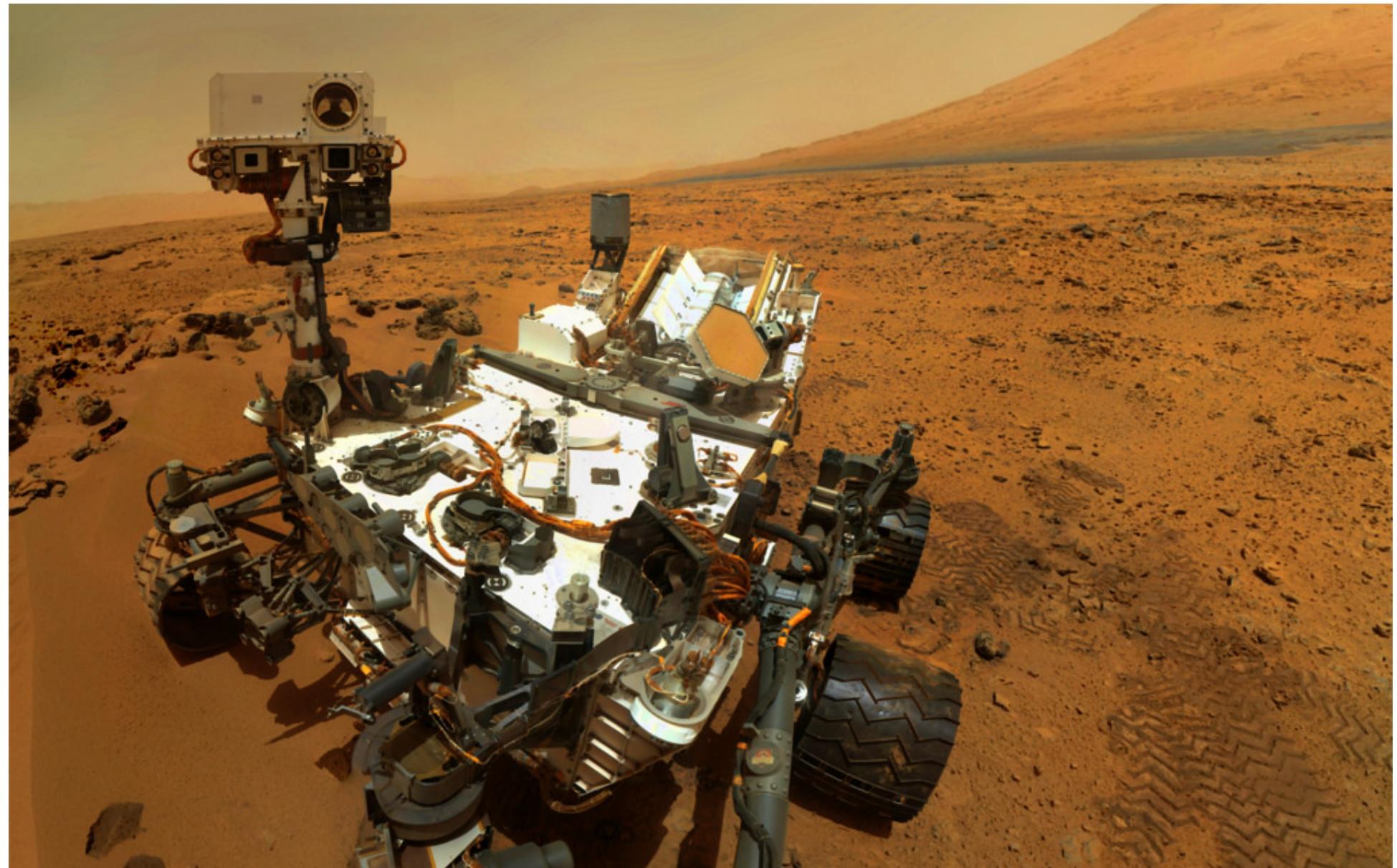
개발자가 갖춰야 할 덕목

Geekiness

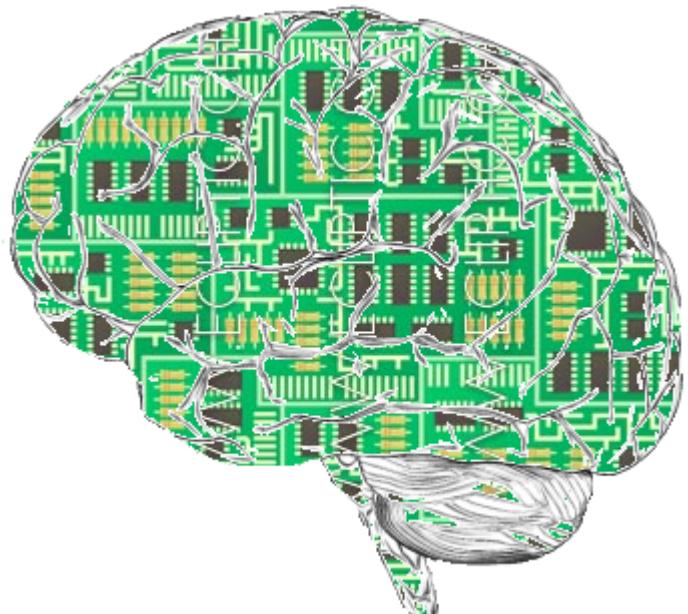


Geek.

Curiosity



Computational Thinking



Queue



Queue

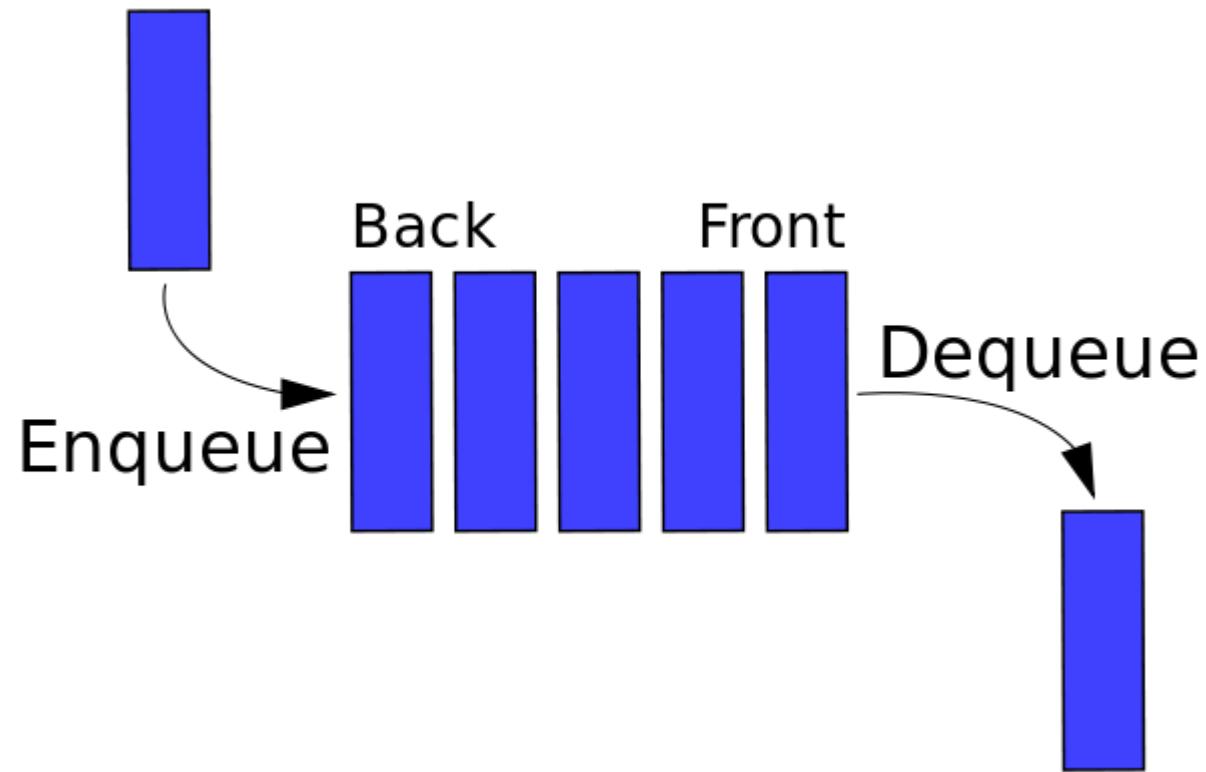
a queue is a particular kind of abstract data type or collection in which the entities in the collection are kept in order.

FIFO

First In First Out

Enqueue & Dequeue

- Enqueue: addition of entities to the rear terminal position
- Dequeue: removal of entities from the front terminal position



Let's Create Queue class

```
function Queue() {  
    //properties, methods  
    var items = [];  
}
```

Enqueue & Dequeue

```
function Queue() {  
    //properties, methods  
    this.enqueue = function(element) {  
        items.push(element);  
    };  
    this.dequeue = function() {  
        return items.shift();  
    };  
}
```

front & isEmpty

```
function Queue() {
    //underneath Enqueue & Dequeue
    ...
    this.front = function() {
        return items[0];
    };
    this.isEmpty = function() {
        return items.length == 0;
    };
}
```

clear & size & print

```
function Queue() {
    //underneath front & isEmpty
    ...
    this.clear = function() {
        items = [];
    };
    this.size = function() {
        return items.length;
    };
    this.print = function() {
        console.log(items.toString());
    };
}
```

Let's Enqueue with Queue class

```
> var queue = new Queue();
> console.log(queue.isEmpty());

> queue.enqueue("Fast");
> queue.enqueue("Campus");
> queue.enqueue("School");

> queue.print();
> console.log(queue.size());
> console.log(queue.isEmpty());
```

Let's Dequeue with Queue class

```
> queue.dequeue();
> queue.dequeue();
> queue.print();
```

collaborate with git flow

Method 2: Fork and Merge

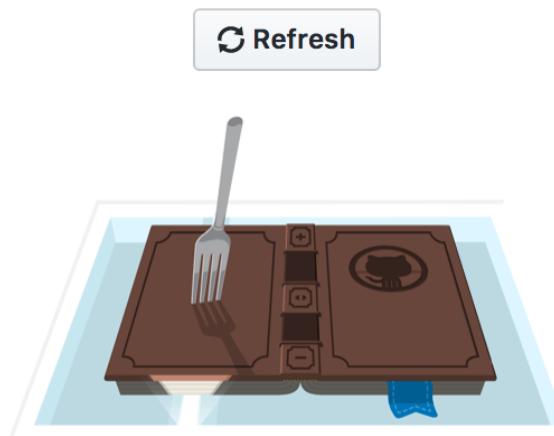


CS ▾

Fork and Merge

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

It should only take a few seconds.



Fork and Merge

 [ulgooon / 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.

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 + 

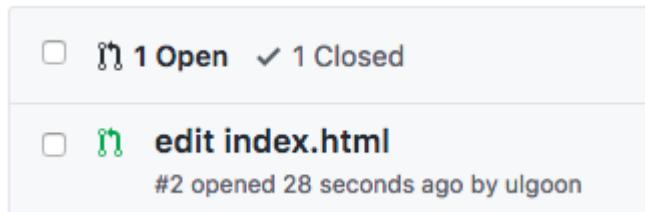
add header, footer tag

 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

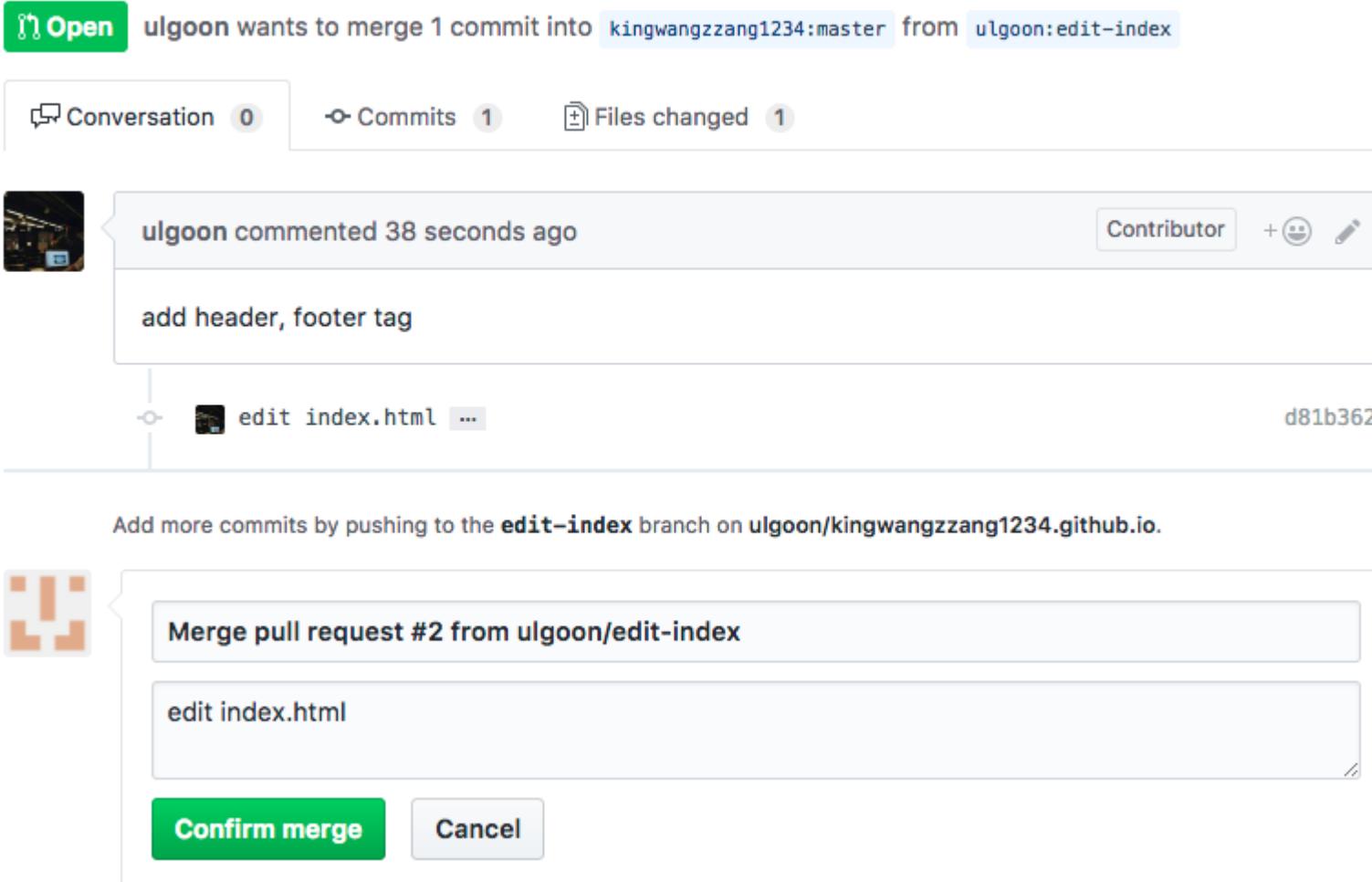
edit index.html ... d81b362

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



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 + 🎉

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 is a "Contributor" button with a plus sign and a smiley face, and a pencil icon. Below the comment is 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

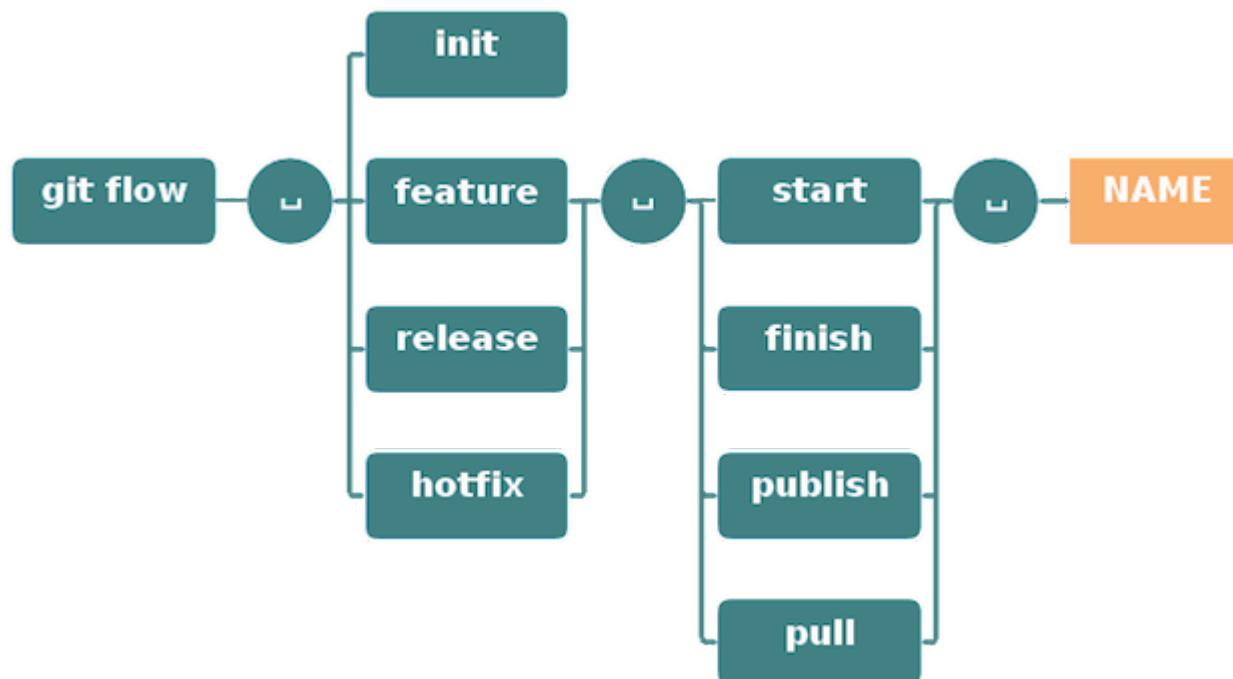
continuous pull

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

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

use git flow easily!

Link



git flow practice

master -> develop -> features -> develop -> release -> master

with Data Structure

Create Stack(), QueueWithStack()

- 3 members 1 repo
- pull 받아 사용해야 하며, 피처 브랜치에서 작업해야 합니다.
- Requirement
 - stack
 - push, pop
 - peek, isEmpty
 - size, print
 - queue with stack
 - enqueue, dequeue

Create Queue with 2 Stacks

```
function Queue_with_stack() {
    var inBox = [];
    var outBox = [];

    this.enqueue = function(num) {
        inBox.push(num);
    };

    this.dequeue = function() {
        if (outBox.length > 0) {
            return outBox.pop();
        }

        while(inBox.length > 1) {
            outBox.push(inBox.pop());
        }
    }

    return inBox.pop();
}
}
```

Gulp



gulp.js

The streaming build system

Gulp is..



Task runner

- 매우 귀찮은 루틴한 작업들을 자동화 할 수 있는 툴
- 현재 2735 + a 개의 패키지가 존재
 - 따라서 필요한 기능을 골라 설치할 필요가 있음!!

task flow

코드작성 – JS test(jshint) – JS Minify – JS Merge(concat) – CSS
Minify – CSS Merge – 결과물

```
$ npm install gulp --global  
$ npm install gulp --save-dev
```

```
$ touch gulpfile.js

var gulp = require("gulp");

//hello라는 gulp task를 등록
gulp.task("hello", function () {
    return console.log("hello gulpworld");
});

$ gulp hello
```

gulp 기본 문법

- `gulp.task` : gulp의 작업단위
- `gulp.src` : gulp 실행의 대상
- `gulp.dest` : gulp 실행 후 목적지
- `gulp.watch` : 변화 감지 후 자동 실행

기본값 설정하기

```
$ gulpfile.js

var gulp = require("gulp");

//hello라는 gulp task를 등록
gulp.task("hello", function () {
    return console.log("hello gulpworld");
});

gulp.task("default", ["hello"]);

$ gulp
```

우선순위 설정하기

```
$touch gulpfile.js

var gulp = require("gulp");

//hello라는 gulp task를 등록
gulp.task("hello", function () {
    return console.log("hello");
});

gulp.task("gulpworld", ["hello"], function () {
    return console.log("gulpworld");
});

gulp.task("default", ["gulpworld"]);

$ gulp
```

자주쓰는 목적지 설정하기

```
var publicPath = {  
    src : './public/src/',  
    dest : './public/dist/'  
};
```

uglify(gulp-uglify) : js uglify

```
gulp.task("uglify", function(){
    pump([
        gulp.src(publicPath.src + 'js/uglify.js'),
        uglify(),
        gulp.dest(publicPath.dest + 'js/')
    ]);
});
```

gulp-concat : js concatenate

```
gulp.task("concat", function(){
    pump([
        gulp.src([publicPath.src + 'js/concat1.js', publicPath.src + 'js/concat2.js'],
            {buffer: false})
            .pipe(concat('concatenated.js')),
        gulp.dest(publicPath.dest + 'js/')
    ]);
});
```

gulp-imagemin : image minify

```
gulp.task("imagemin", function(){
  pump([
    gulp.src(publicPath.src + 'img/*.jpg'),
    imagemin(),
    gulp.dest(publicPath.dest + 'img/')
  ]);
});
```

css minify(gulp-clean-css) : css minify

```
gulp.task("cleancss", function(){
    pump([
        gulp.src(publicPath.src + 'css/minify.css'),
        cleancss(),
        gulp.dest(publicPath.dest + 'css/')
    ]);
});
```

gulp-sass : convert .scss to .css

```
gulp.task("sass", function(){
  pump([
    gulp.src(publicPath.src + 'sass/*.scss'),
    sass().on('error', sass.logError),
    gulp.dest(publicPath.dest + 'css/')
  ]);
});
```

gulp-concat-css : concatenate css files

```
gulp.task("concatcss", function(){
  pump([
    gulp.src([publicPath.src + 'css/concat1.css', pu
      concat('concatenated.css'),
      gulp.dest(publicPath.dest + 'css/')
  ]);
});
```

clean(del)

```
gulp.task("clean", function(){
    return del.sync([publicPath.dest + 'js/*.js', publicPath
});
```

watch

```
gulp.task("watch", function(){
    gulp.watch("public/src/*.js", ["uglify"]);
});

gulp.task("default", ["uglify", "watch"]);
```