Fastcampus

Web Programming & Frontend Dev SCHOOL

Web_socket, Data Structure, gulp

Websocket

Websocket

> 웹사이트가 사용자와 상호작용하기 위해 만들어진 기술

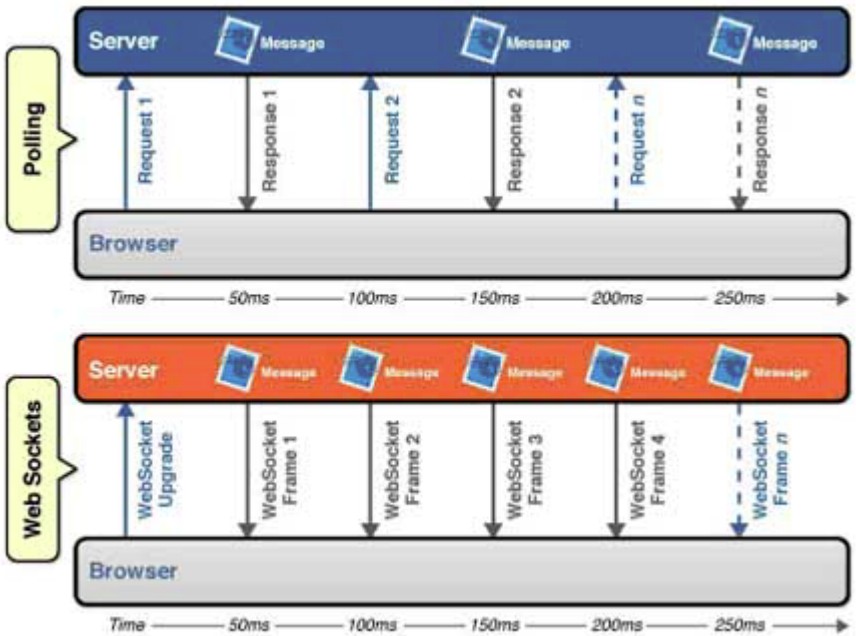
W3C가 API를 관리

port:80, HTTP1.1

Before Websocket

- HTTP Request, Response
- Hidden Frame
- Long Polling

Polling vs Websocket



Differences between Socket, Websocket

Socket - HTTP run over TCP/IP

Websocket - run from web browser

Socket.io

> browser 와 상관없이 js를 이용해 실시간 통신을 지원

```
npm install --save socket.io
```

- 브라우저와 웹 서버의 종류와 버전을 분석해 가장 적절한 기술로 통신
- WebSocket, FlashSocket, AJAX Long Polling, AJAX Multi part Streaming, IFrame, JSONP Polling을 추상화한 기술
- 일관성있는 문법과 API로 개발 가능
- IE6 부터 최신 브라우저까지 지원

chat.ejs

```
<style>
        * { margin: 0; padding: 0; box-sizing: border-box; }
        body { font: 13px Helvetica, Arial; }
        form { background: #000; padding: 3px; position: fixed;
        form input { border: 0; padding: 10px; width: 90%; margi
        form button { width: 9%; background: rgb(130, 224, 255);
        #messages { list-style-type: none; margin: 0; padding: 0
        #messages li { padding: 5px 10px; }
        #messages li:nth-child(odd) { background: #eee; }
</style>

<ul id="messages"></ul>
<form action="">
        <input id="m" autocomplete="off" /><button>Send</button>
</form>
```

connection

```
var http = require("http").Server(app);
var io = require("socket.io")(http);

io.on("connection", function(socket){
        console.log("a user connected");
});

http.listen(8080, function(){
        console.log("Chat Server is running at localhost:8080");
});
```

```
<script src="/socket.io/socket.io.js"></script>
<script>
        var socket = io();
</script>
```

disconnect

```
socket.on("disconnect", function(){
        console.log("user disconnected");
});
```

emit message

```
socket.on("chat message", function(msg){
        console.log("message: " + msg);
```

```
<script src="http://code.jquery.com/jquery-1.11.1.js"></script>

var socket = io();
$("form").submit(function(){
        socket.emit("chat message", $("#m").val());
        $("#m").val("");
        return false;
});
```

broadcast

```
io.emit("chat message", msg);
```

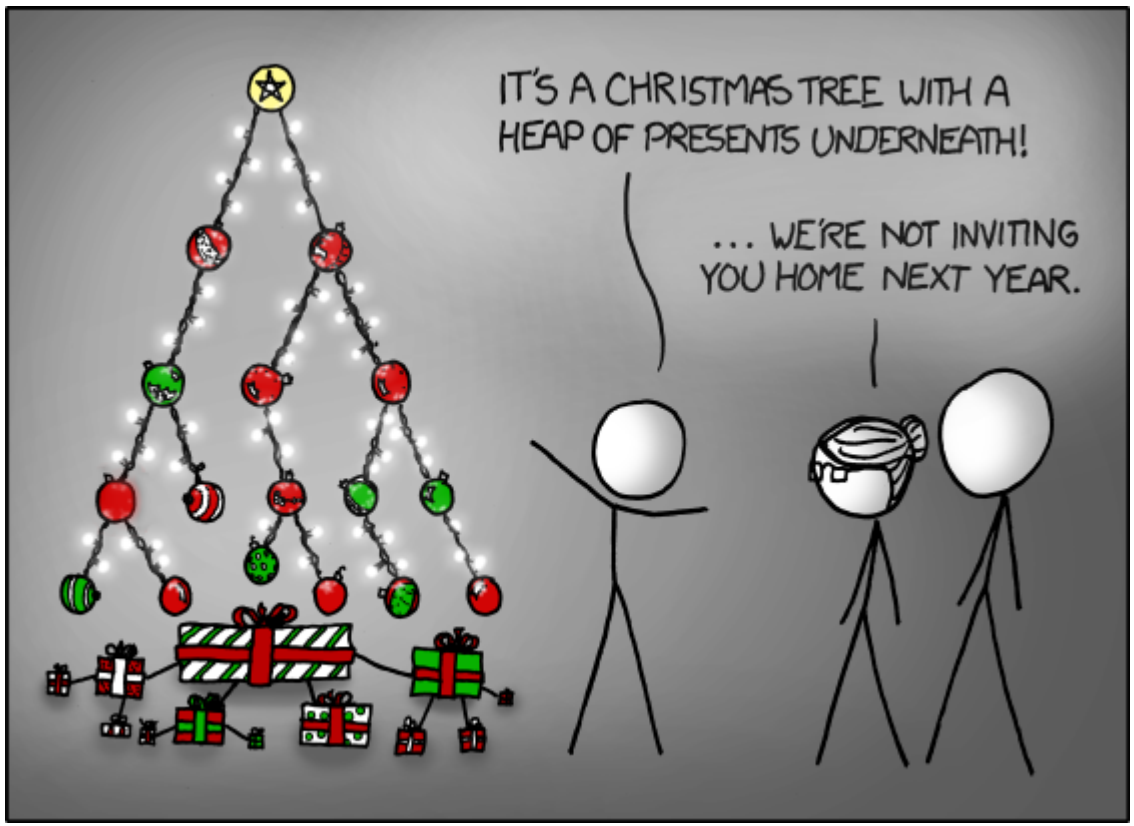
```
socket.on("chat message", function(msg){
        $("#messages").append($("<li>").text(msg));
});
```

Data Structure

Data Structure

> Data structure is a particular way of organizing data in a computer so that it can be used efficiently.

So, Data Structure is..



Data Structures in Web Development

Array & Hash(Dictionary) - indexing post

```
in RDB
[articleId, title, body, userId, view]
```

```
[{
        userId: 1,
        articleId: 1,
        view: 100,
        title: "sunt aut facere repellat provident occaecati exc
        body: "quia et suscipit suscipit recusandae consequuntur
},
...

]
```

Data Structures in Web Development

Tree - DOM rendering performance, reply

```
<html>
<head></head>
<body>
<h1></h1>
<p></p>
</body>
</html>
```

Data Structures in Web Development

- Binary Tree Search
  - Queue(BFS, Breadth First Search)
  - Stack(DFS, Depth First Search)

Stack

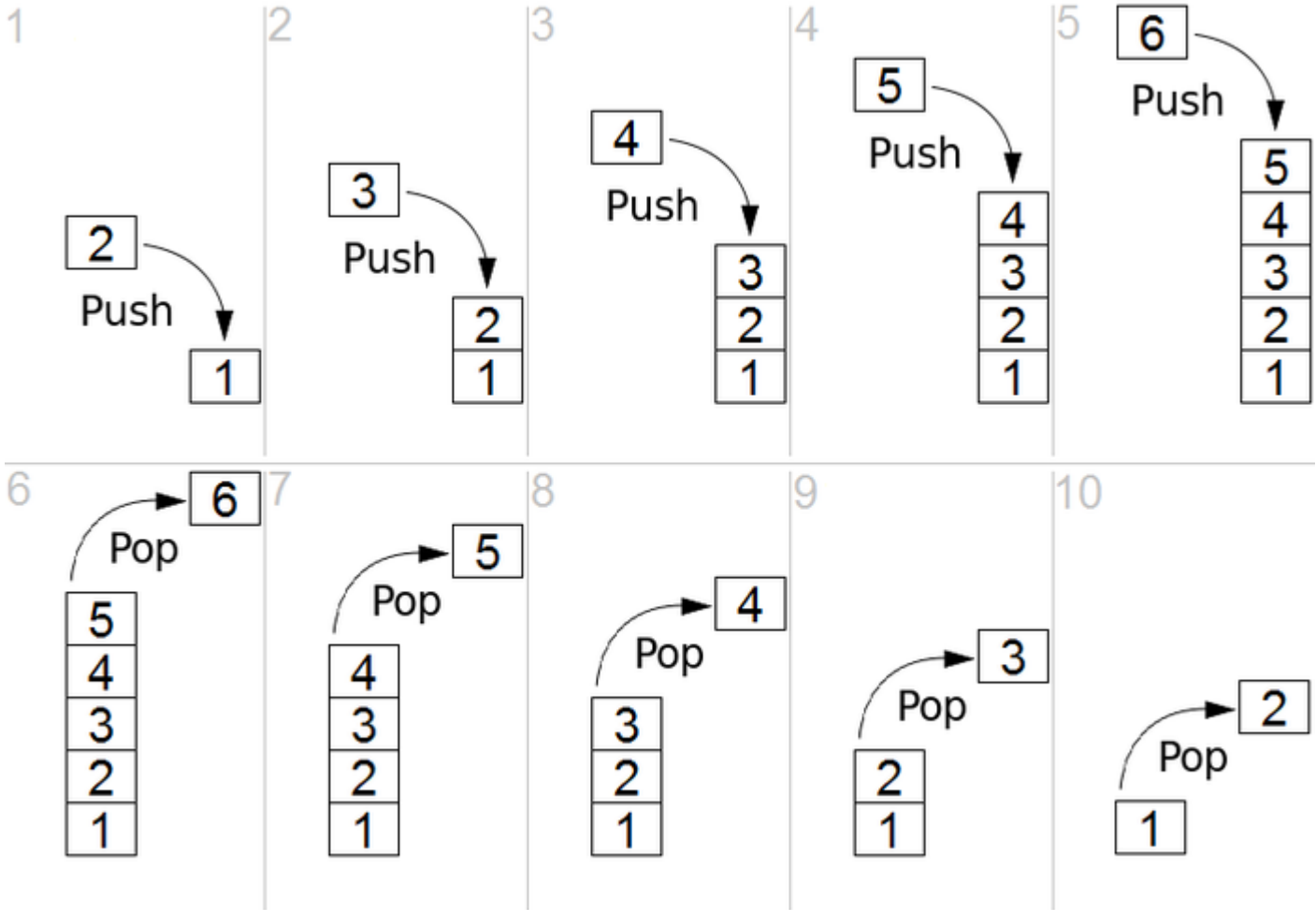
Stack

> A stack is an abstract data type that serves as a collection of elements, with two principal operations.

- push: which adds an element to the collection
- pop: which removes the most recently added element that was not yet removed

LIFO

> Last In, First Out


1
2
Push
1
2
3
Push
2
1
3
4
Push
3
2
1
4
5
Push
4
3
2
1
5
6
Push
5
4
3
2
1
6
Pop
6
5
4
3
2
1
7
Pop
5
4
3
2
1
8
Pop
4
3
2
1
9
Pop
3
2
1
10
Pop
2
1

Queue

Queue

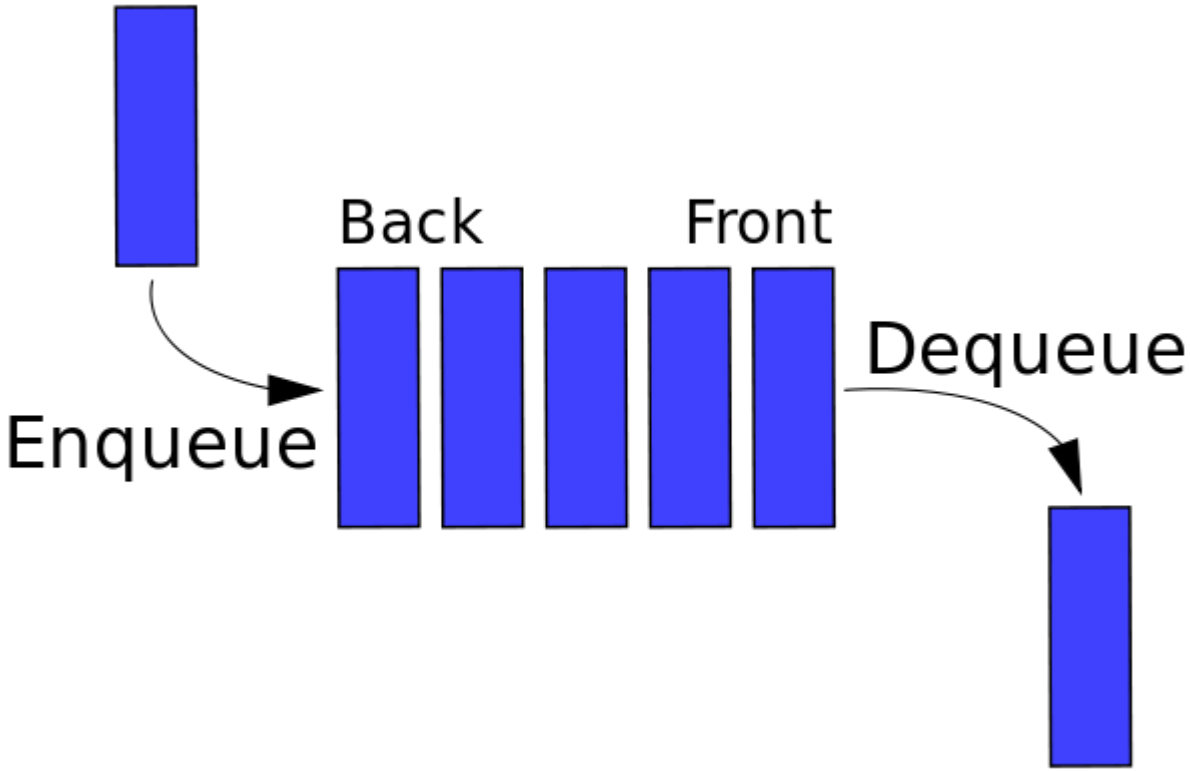
> a queue is a particular kind of abstract data type or collection in which the entities in the collection are kept in order.

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



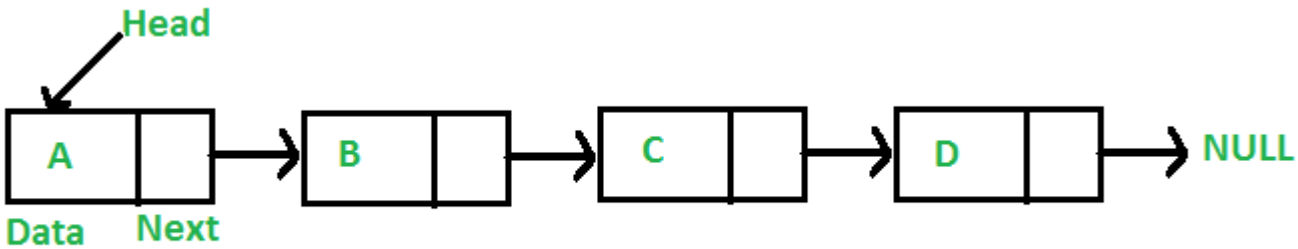
Linked List

Linked List

> A linked list is a linear collection of data elements, in which linear order is not given by their physical placement in memory.

Linked List

- Can be used to store linear data of simlar types.

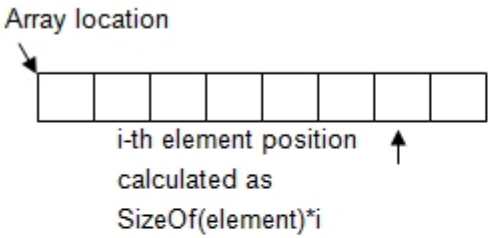


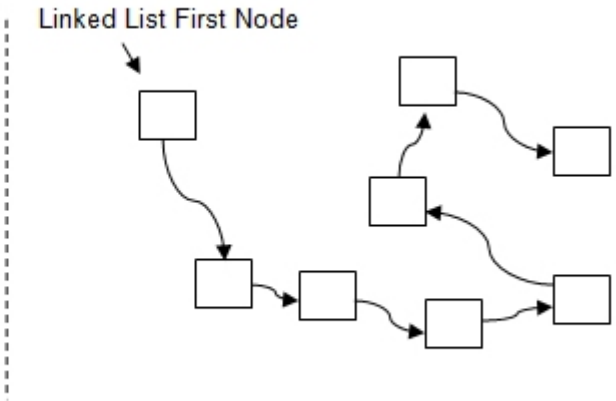
Array를 놔두고 굳이 왜???

Array

```
var myArray = [];
```

Array vs Linked List





Array vs Linked List

Array	vs	Linked List
Fixed	Size	Dynamic
Hard	Insert	Easy
Hard	Deletion	Easy
Allowed	Random Access	Not allowed
doesn't need	Extra memory space	required

Array vs Linked List

Array	vs	Linked List
o(1)	access	o(n)
o(n)	search	o(n)
o(n)	insert	o(1)
o(n)	remove	o(1)

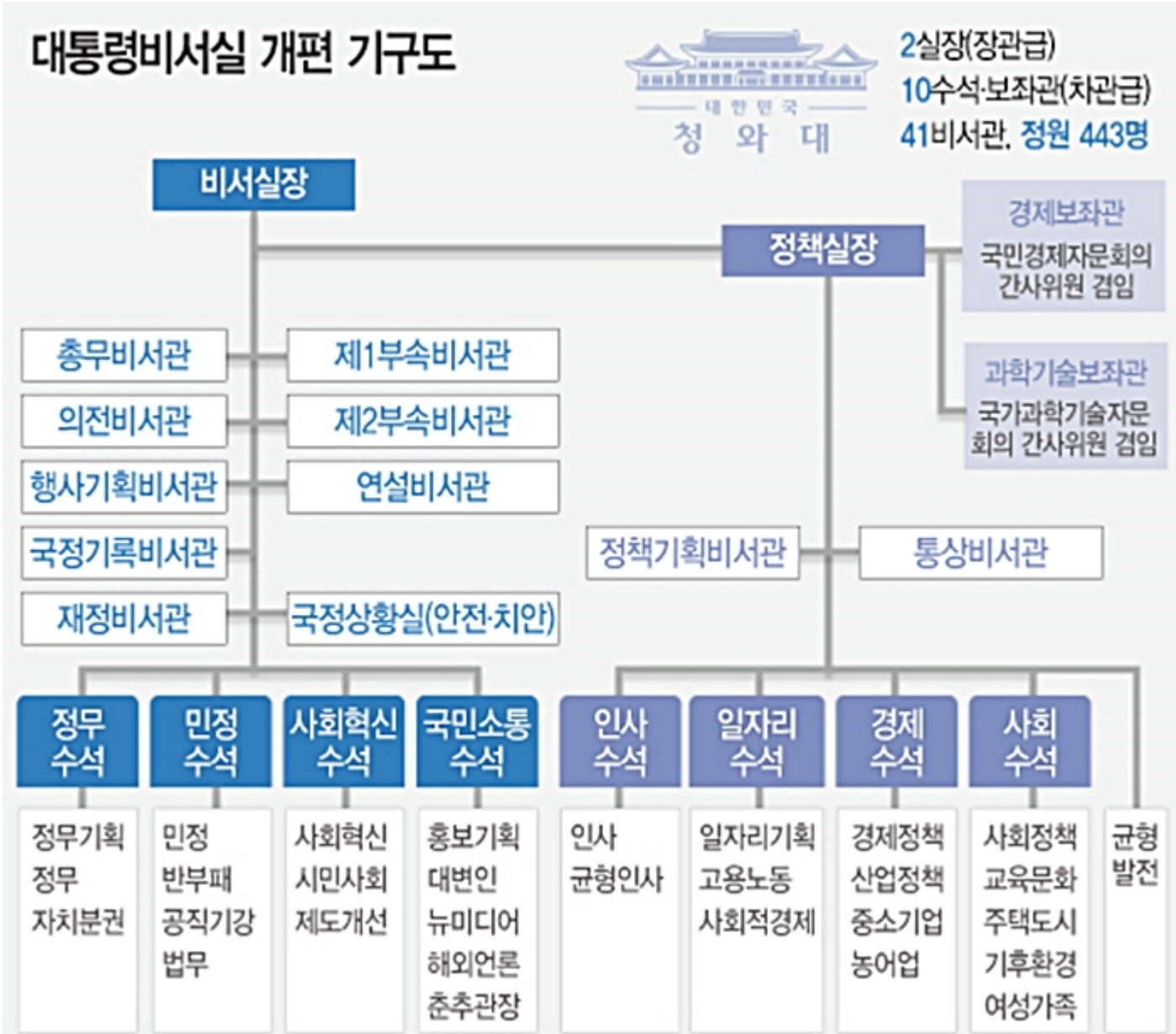
Tree

Tree

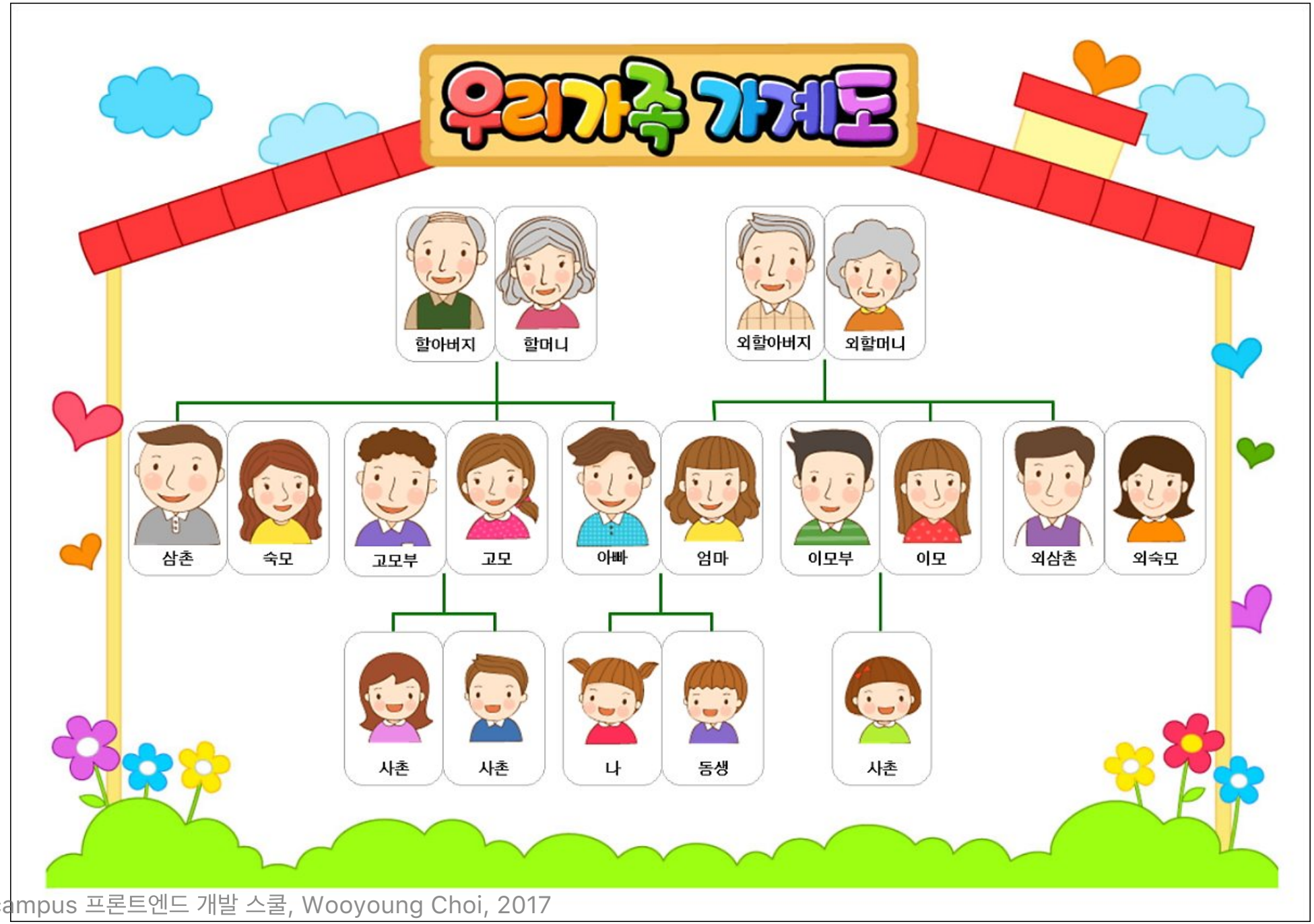
> A tree is an abstract model of a hierarchical structure.

- hierarchical: arranged in order of rank.

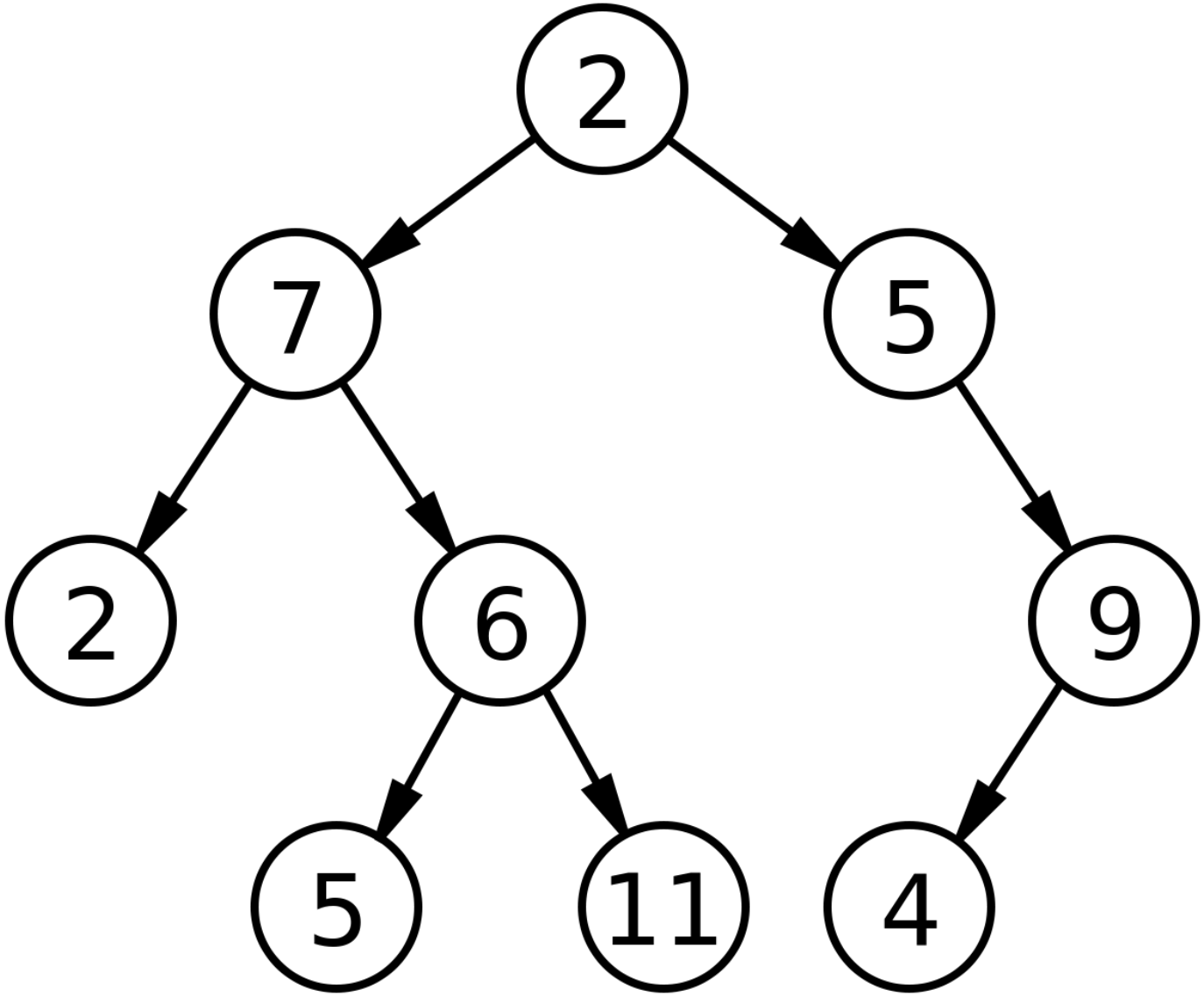
Tree


대통령비서실 개편 기구도
대 한 민 국
청 와 대
2실장(장관급)
10수석·보좌관(차관급)
41비서관, 정원 443명
비서실장
정책실장
경제보좌관
국민경제자문회의 간사위원 겸임
과학기술보좌관
국가과학기술자문회의 간사위원 겸임
총무비서관
제1부속비서관
의전비서관
제2부속비서관
행사기획비서관
연설비서관
국정기록비서관
재정비서관
국정상황실(안전·치안)
정책기획비서관
통상비서관
정무 수석
민정 수석
사회혁신 수석
국민소통 수석
인사 수석
일자리 수석
경제 수석
사회 수석
정무기획
정무
자치분권
민정
반부패
공직기강
법무
사회혁신
시민사회
제도개선
홍보기획
대변인
뉴미디어
해외언론
춘추관장
인사
균형인사
일자리기획
고용노동
사회적경제
경제정책
산업정책
중소기업
농어업
사회정책
교육문화
주택도시
기후환경
여성가족
균형발전

Tree


우리가족 가계도
할아버지
할머니
외할아버지
외할머니
삼촌
숙모
고모부
고모
아빠
엄마
이모부
이모
외삼촌
외숙모
사촌
사촌
나
동생
사촌

Tree



Tree

- root: 2
- level: (0 ~ 3)
- child of 2: 7,5
- subtree: 6,5,11
- Node: (9)
- edge: (8)

Gulp

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', publ
                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

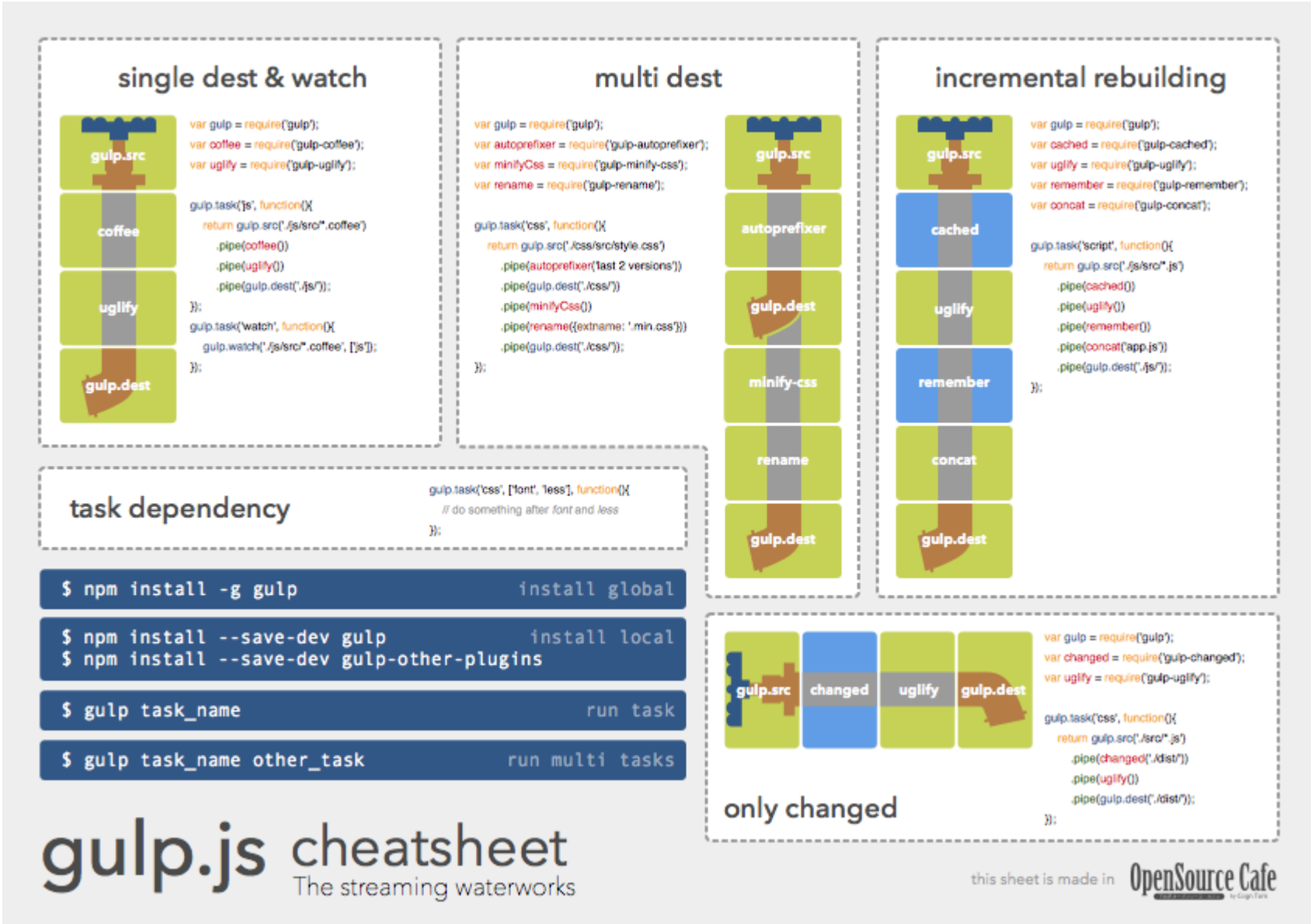
```
gulp.task("watch", function(){
        gulp.watch("public/src/*.js", ["uglify"]);
});

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

watch

```
gulp.task("watch", function(){
        gulp.watch(publicPath.src + 'js/*.js', ["uglify", "conca
        gulp.watch(publicPath.src + 'css/*.css', ["cleancss", "c
        gulp.watch(publicPath.src + 'img/*.jpg', ["imagemin"]),
        gulp.watch(publicPath.src + 'sass/*.scss', ["sass"])
 +});
```

이외에도..


single dest & watch
gulp.src
coffee
uglify
gulp.dest
var gulp = require('gulp');
var coffee = require('gulp-coffee');
var uglify = require('gulp-uglify');
gulp.task('js', function(){
return gulp.src('./js/src/*.coffee')
.pipe(coffee())
.pipe(uglify())
.pipe(gulp.dest('./js/'));
});
gulp.task('watch', function(){
gulp.watch('./js/src/*.coffee', ['js']);
});
multi dest
var gulp = require('gulp');
var autoprefixer = require('gulp-autoprefixer');
var minifyCss = require('gulp-minify-css');
var rename = require('gulp-rename');
gulp.task('css', function(){
return gulp.src('./css/src/style.css')
.pipe(autoprefixer('last 2 versions'))
.pipe(gulp.dest('./css/'))
.pipe(minifyCss())
.pipe(rename({extname: '.min.css'}))
.pipe(gulp.dest('./css/'));
});
gulp.src
autoprefixer
gulp.dest
minify-css
rename
gulp.dest
incremental rebuilding
gulp.src
cached
uglify
remember
concat
gulp.dest
var gulp = require('gulp');
var cached = require('gulp-cached');
var uglify = require('gulp-uglify');
var remember = require('gulp-remember');
var concat = require('gulp-concat');
gulp.task('script', function(){
return gulp.src('./js/src/*.js')
.pipe(cached())
.pipe(uglify())
.pipe(remember())
.pipe(concat('app.js'))
.pipe(gulp.dest('./js/'));
});
task dependency
gulp.task('css', ['font', 'less'], function(){
// do something after font and less
});
$ npm install -g gulp
install global
$ npm install --save-dev gulp
$ npm install --save-dev gulp-other-plugins
install local
$ gulp task_name
run task
$ gulp task_name other_task
run multi tasks
gulp.src
changed
uglify
gulp.dest
var gulp = require('gulp');
var changed = require('gulp-changed');
var uglify = require('gulp-uglify');
gulp.task('css', function(){
return gulp.src('./src/*.js')
.pipe(changed('./dist/'))
.pipe(uglify())
.pipe(gulp.dest('./dist/'));
});
only changed
gulp.js cheatsheet
The streaming waterworks
this sheet is made in OpenSource Cafe