

Back-End Programming for Front-End Developers

Day5

프론트엔드 개발 SCHOOL 2기, FASTCAMPUS

2016.11.25

Todo

Sass compile

완벽한 예제

heroku deploy

custom domain

종강파티

AMP

클라이언트 개발자라면 언젠가는 하게 될 프로젝트

공식 홈페이지

[Google Developers Korea blog - AMPlify](#)

gulp

- 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'
        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',
            concat('concatenated.css'),
            gulp.dest(publicPath.dest + 'css/')
        ]
    ]);
});
```


clean(del)

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

watch

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

watch

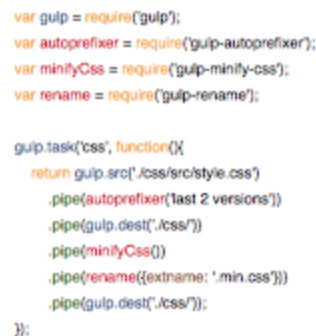
```
gulp.task("watch", function(){
    gulp.watch(publicPath.src + 'js/*.js', ["uglify",
    gulp.watch(publicPath.src + 'css/*.css', ["cleancss",
    gulp.watch(publicPath.src + 'img/*.jpg', ["imagemin",
    gulp.watch(publicPath.src + 'sass/*.scss', ["sass"]);
    +});
```

이외에도..

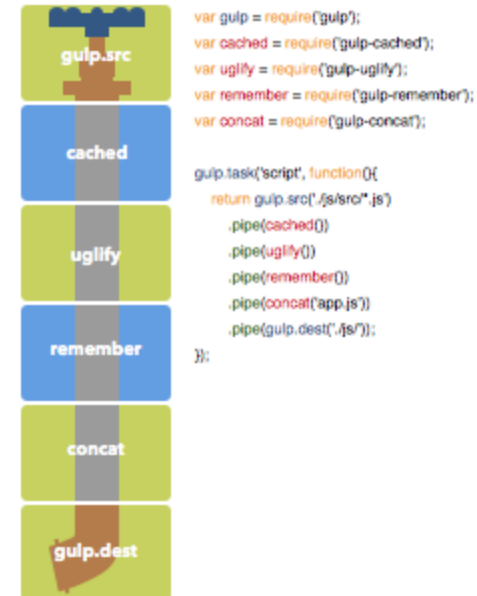
single dest & watch



multi dest



incremental rebuilding



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 install local

\$ npm install --save-dev gulp-other-plugins

\$ gulp task_name run task

\$ gulp task_name other_task run multi tasks

gulp.js

cheatsheet
The streaming waterworks



only changed

this sheet is made in **OpenSource Cafe**
for Open Source

Deploy with Heroku

Cloud Server

- AWS
- Azure
- Google cloud platform
- [Heroku](#)

Deploy with Heroku

[heroku cli download link](#)

```
$ brew install heroku-toolbelt  
$ heroku update  
  
$ heroku login
```

Heroku Deploy

```
$ heroku apps:create { nickname }  
$ heroku apps  
  
$ heroku apps:delete { nickname }
```

```
$ heroku git:remote -a { nickname }  
$ git push heroku master  
  
$ heroku ps:scale web=1  
$ heroku open
```

```
{ nickname }.herokuapp.com 접속
```

Heroku Operation

```
// show server status
```

```
$ heroku ps
```

```
// view logs
```

```
$ heroku logs -tail
```

```
// shutdown
```

```
$ heroku ps:scale web=0
```


챌린지!!

40라인으로 슬랙봇만들기 를 이해하고 커스터마이즈 할 수 있다면 여러분은 node.js express.js heroku git REST api를 이해하셨습니다.