



프론트엔드 개발도구 태스크러너? 빌드도구 ! GULP를 알아보자!

작성자 : 아라한사

2015-05-25

페북 : <https://fb.com/me.adunhansa>

트위터 : <https://twitter.com/arahansa>

블로그: <http://adunhansa.tistory.com/>

사이트 : <http://arahansa.com>





닉 : **아라한사**

주력&관심 기술 : Java, Spring, ORM, Go

블로그 <http://arahansa.com>

평소 정리를 즐겨합니다. 한 때 동영상강좌도

만들다보니.. 비실명과 만화캐릭터를 쓰네요

양해 부탁드립니다 ㅎㅎ

1. **작성자** 소개

- 코딩하는 동안 마음의 평화를 얻고, 온갖 108버그의 번뇌에서 벗어나고 싶은 부처님 오신 날을 맞이하여 프론트엔드 태스크 관리 도구 **gulp**를 소개합니다!

미혹을 버리고
좋은 도구를 찾는 이들에게
번뇌의 시간이 없으리니..



늘어만 가는 프론트엔드도구들과 프레임워크..
그냥 듣기만 한 것들이라도 적어봅니다..-_-

- grunt, gulp
- bower , yeoman
- underscore, phantom.js , Casper.JS
- Angular.js, React.js, ext.js
- meteor.js, backbone.js
- js lint
- selenium
- require.js
- less , sass



-_-예.. 쯔긴 쯔은 거일텐데.. 정말 많습니다.

(짤방: 이말년 씨리즈 : <https://mirror.enha.kr/wiki/고만해, 미친놈들아!>)

- 프론트엔드도구들과 js프레임워크들..하나씩
알아보고자 합니다.
같이 하나하나씩 익혀봅시다.



이제는 피할 수 없다. 프론트엔드 도구 ! alaboza.

(이말년 짤방 : 좌절위키 참조 <http://orzwiki.com/>분류:분류)

오늘은, Gulp 부터 알아보고자 합니다.
프론트엔드 도구들은 node.js의 npm으로 많이 받아옵니다.
미리 설치하셨다고 가정합니다.



참고 사이트

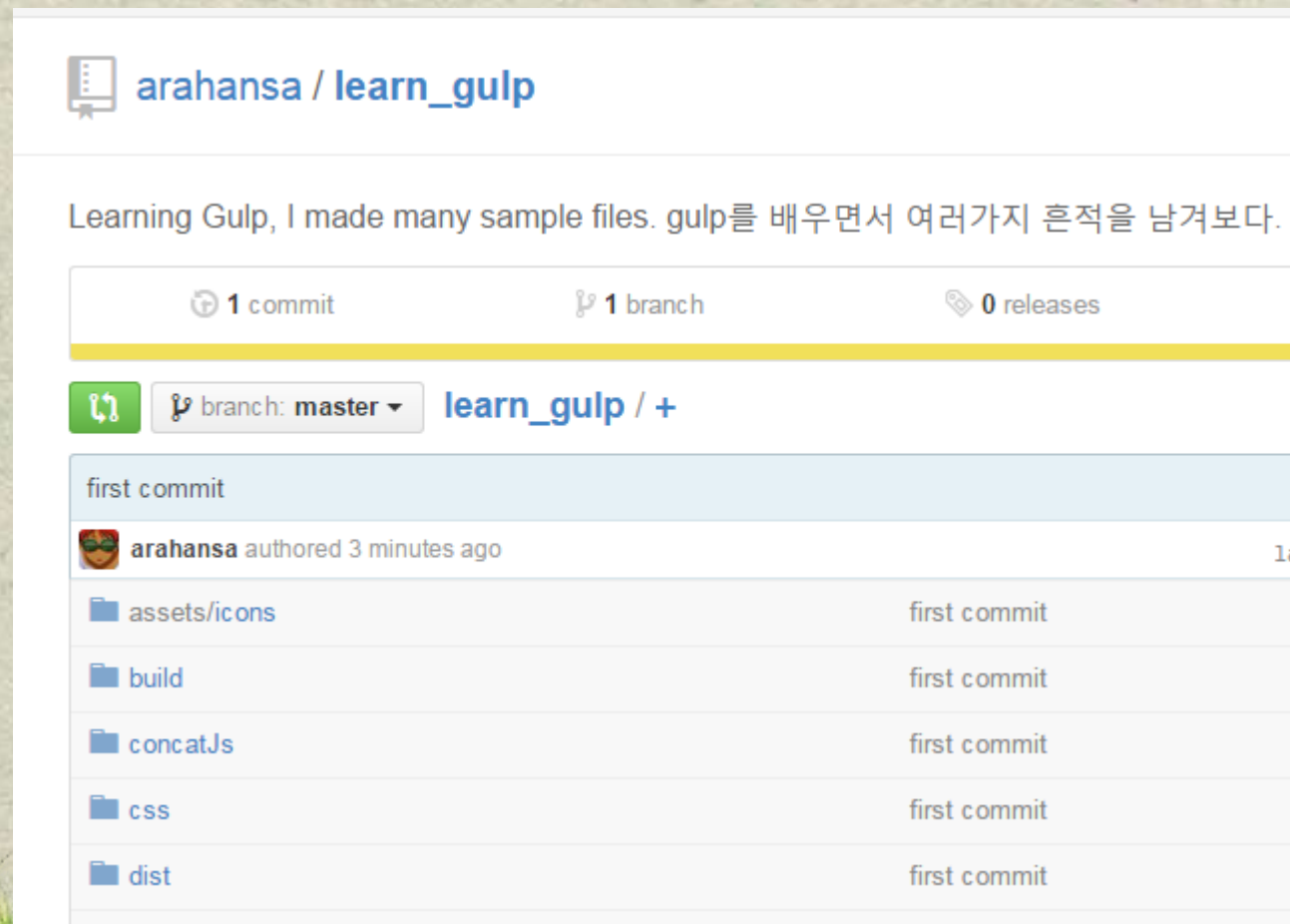
<https://github.com/gulpjs/gulp>

<http://gulpjs.com/>

<https://www.npmjs.com/>

- 여러 깃헙에서 이것저것 소스품앗이를 다니면서 폴더설정등을 가능한 그대로 유지하다보면서 하다보니 **폴더가 좀 번잡**합니다-_-; (하지만 개발자분들은 똑똑하시니 다 이해하시리라;;믿으며;;) 각각의 플러그인 기능에 집중하며 이것을 통틀어서 실제 서버를 돌리기 위한 사례는 여기선 다루지 않습니다.
- 대부분 간단한 **깃헙에서 참조한 소스**로 이뤄져있습니다. 보기에다 사실 간단합니다. (저야..뭐.. 소스 약간 안 맞는거 조금 삽질하고, 사진 땀질 좀 했지만요..생색;;)
- 아..아무튼 개발자는 이상하게 매일 바뀝니다. 그래서 요약해보자면, 초반에만 gulp설명을 좀 읽어보시고, task 어떻게 돌리는 지 보시면 될 것같구요. 테스트부분 이전에 나오는 **watch랑 brower-sync**는 재미있는 기능입니다. 이부분만 조금 눈여겨 보시면 될 것같습니다.
- **js 테스트 부분은 저도 익숙하지는 않고** 한 번 해보면서 이게 맞는지 읽으시는 분께 물어보는느낌으로 적어봤습니다. 사실 gulp도 이번이 처음 입문하면서 해보는 거라.. 틀릴 수도 있는 부분이 많습니다.^^;

- 본 슬라이드는 깃헙 소스를 함께 제공합니다.
https://github.com/arahansa/learn_gulp
- 다운을 받으시고 npm install 을 치신다음에 잘 동작하는지 확인을 해보시면 될 것입니다.



제공하는 깃헙 소스

필요성(잡설)

빌드도구 gulp개념?

01. 헬로월드

02. 난독화 uglify

03. 다른 task생성

04. css압축 minify

05. 로깅 util

06. 파일 연결 concat

07. 파일명 변경 rename

08. 작업동시처리sourcemaps

09. 에러처리 plumber

10. 아이콘

11. 벤더프리픽스 자동처리

12.플러그인 의존성관리

13. 이미지 압축

14. sass 컴파일

15. 파일 변경 감지 watch

16. 브라우저 가동 browser-sync★

테스트도구들

17. jslint

18. jasmine

19. jasmine-phantom.js

마치며...

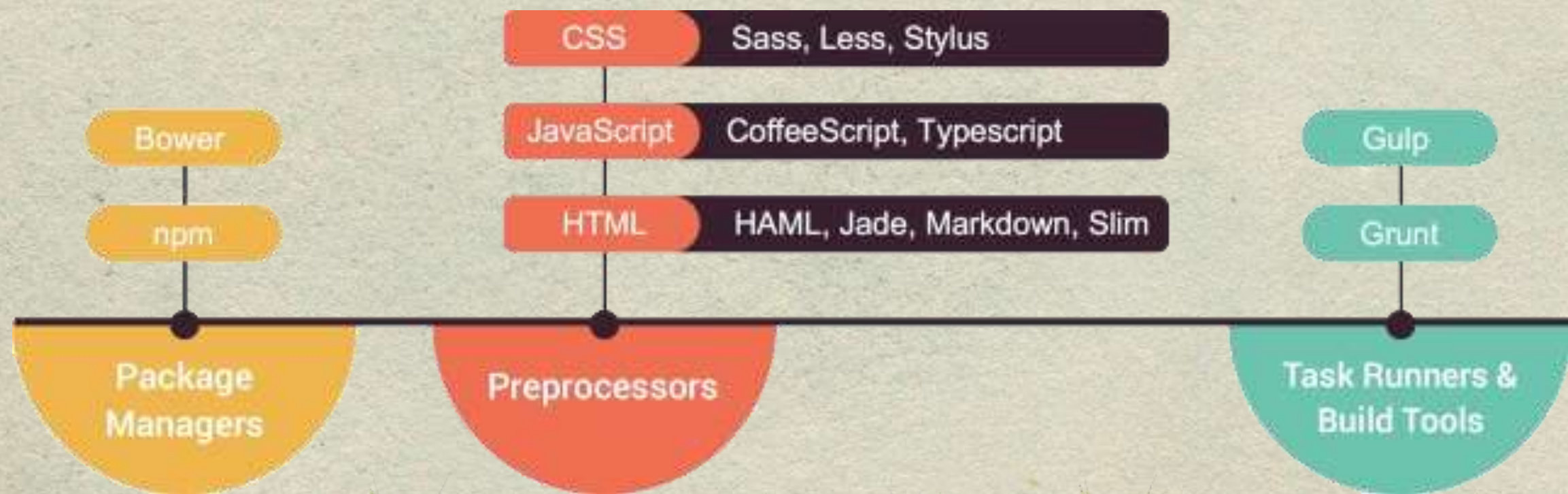
목차 (링크 연결해뒀습니다)



- 주로 백엔드를 공부하면서 처음엔 gulp를 왜 쓰는지 감이 오질않았습니다.
- Grunt부터 보긴 했는데, Grunt는 상대적으로 좀 어려워보이기도 했습니다.
- 흠 gulp 좋긴 좋은데 한마디로 말하기 어렵군요 ㅠㅠ 이론 내용을 잠깐 보실까요

자, 그럼~ gulp의 **필요성!**

- 빌드시스템은 단순히 task runners라 불리며 반복적으로 자동화된 task들의 모음입니다. (하단 링크 사이트 발번역)
- 일반적 사용은 전처리된 CSS, 자바스크립트 컴파일, 파일들 연결, 최소화, 개발빌드등을 하는 일등을 해주면서 패키지매니저와 같은 도구들과 함께 작업을 합니다. gulp를 위한 슬라이드이므로, 패키지매니저와 전처리설명은 생략^^

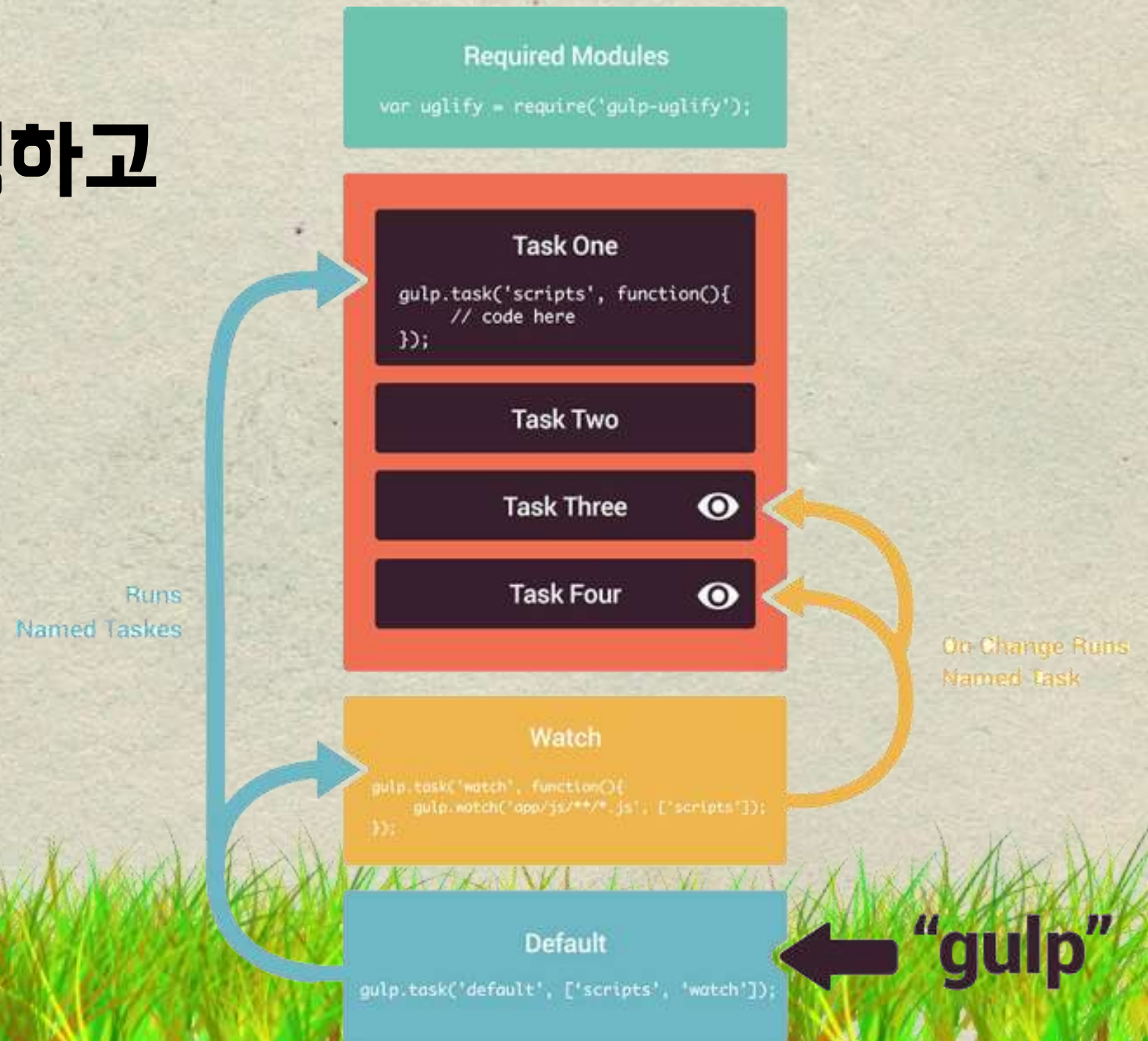


최신의 웹 작업 구성요소들

이미지와 내용 출처 : <http://joellongie.com/gulp-build-system-fundamentals/>

아니.. 이것을 보도록 하지요. **빌드 시스템**이란 무엇인가?

- 다음과 같이 모듈을 require로 받아서
- 여러 Task를 구성하고
- 변화를 감지하고
- default로 여러 태스크를 실행함..



태스크 기반 빌드 도구 Gulp !!

- 이론만 보다보면 어렵습니다; 역시 코딩하면서 봐야;;
호흡 한번 내쉬고 시작해볼까요?



그냥 **코딩**이나 하면서 시작해볼까요! < 14 >

- gulp 폴더를 만들어주겠습니다.

```
Cmder
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> |
```

- npm install --global gulp 로 gulp 설치
- 다음의 소스 코드 입력후 gulffile.js 로 저장

```
var gulp = require('gulp');
gulp.task('default', function() {
  console.log("Gulp야 안녕 ? ^^; ")
});
```

```
gulpfile.js
1 var gulp = require('gulp');
2
3 gulp.task('default', function() {
4   console.log("Gulp야 안녕 ? ^^; ")
5 });
6
```

그리고 gulp를 명령어를 치면 됩니다.

```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> gulp
[21:37:20] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[21:37:20] Starting 'default'...
Gulp야 안녕 ? ^^;
[21:37:20] Finished 'default' after 462 μs
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp>
```

gulp의 인스톨과 01. 헬로월드 !



- gulp에는 task라는 개념이 있습니다.
- node.js 의 require로 gulp모듈을 가져와서 gulp의 task중 기본 default 태스크를 실행시키는 것입니다. 참 쉽죠.

```
gulpfile.js ✕  
1  var gulp = require('gulp');  
2  
3  gulp.task('default', function() {  
4    console.log("Gulp야 안녕 ? ^^; ");  
5  });  
6
```

gulp의 헬로월드도 설명이 필요합니다.

Uglify를 알아보도록 하겠습니다.



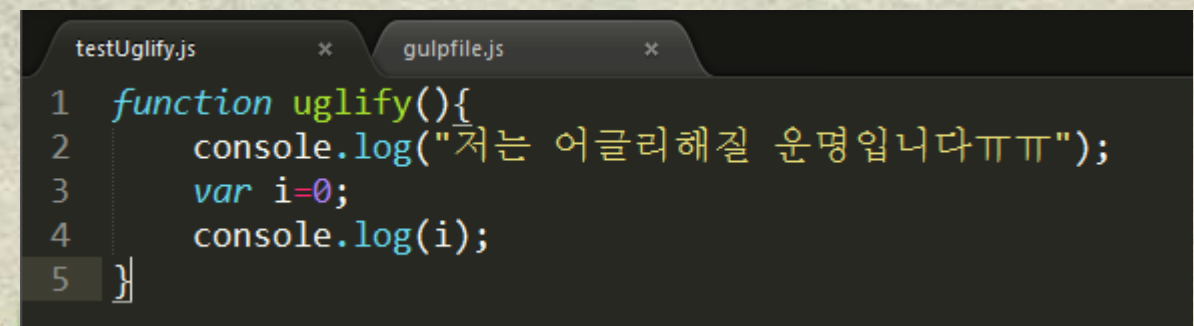
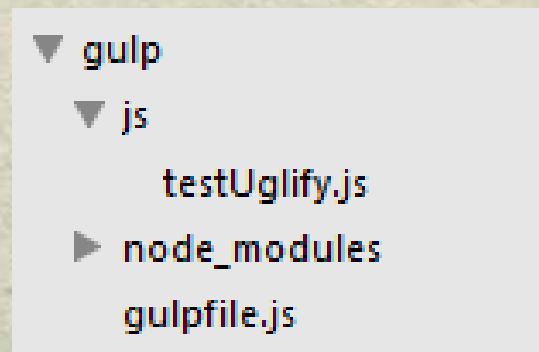
- 소스코드를 난독화시키는 것을 보고 우클릭금지가 연상되면서 치사(?)하군.. 이라고 생각했지만,그게 아닙니다.
- js파일을 압축을 시켜서 빠른 전송을 용이케 합니다.

- 설치 : `npm install --save-dev gulp-uglify`
사이트 : <https://github.com/terinjokes/gulp-uglify>
- 용도 : **난독화**

```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> npm install --save-dev gulp-uglify
gulp-uglify@1.2.0 node_modules\gulp-uglify
├── deap@1.0.0
├── through2@0.6.5 (xtend@4.0.0, readable-stream@1.0.33)
├── vinyl-sourcemaps-apply@0.1.4 (source-map@0.1.43)
├── uglify-js@2.4.19 (uglify-to-browserify@1.0.2, async@0.2.10, source-map@0.1.34, yargs@3.5.
├── gulp-util@3.0.4 (array-differ@1.0.0, object-assign@2.0.0, beeper@1.0.0, array-uniq@1.0.2,
├── replace-ext@0.0.1, minimist@1.1.1, multipipe@0.1.2, vinyl@0.4.6, chalk@1.0.0, dateformat@1
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp>
```

02. **uglify** - 난독화하기

- uglify를 위해 js폴더를 만들어주시고 다음과 같이 소스코드를 적어주겠습니다. 뭐.. 변환될 것이므로 대충 적어줍니다.



- 그럼 이제 Gulp 파일을 만져주겠습니다.
우선 기본 task안에 uglify할 거리를 만들어주겠습니다.

.pipe로 작업을 연결하면서 .dest로 목적지를 설정합니다.

```
var uglify = require('gulp-uglify');  
gulp.src('js/*.js')  
.pipe(plumber())  
.pipe(uglify())  
.pipe(gulp.dest('build/js'));
```

```
gulpfile.js  ×  testUglify.js  ×  
1  var gulp = require('gulp');  
2  var uglify = require('gulp-uglify');  
3  
4  gulp.task('default', function() {  
5    console.log("Gulp야 안녕 ? ^^; ");  
6    gulp.src('js/*.js')  
7      .pipe(uglify())  
8      .pipe(gulp.dest('build/js'));  
9  });  
10
```


- 기본 명령어 gulp 를 실행하면 아까의 설정에서 지정한 build/js 폴더내에 압축된 파일이 생깁니다.

```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> gulp
[22:01:03] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[22:01:03] Starting 'default'...
Gulp야 안녕 ? ^^;
[22:01:03] Finished 'default' after 8.79 ms
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp>
```



```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> cd build/js
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\build\js> dir

디렉터리: C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\build\js

Mode                LastWriteTime         Length Name
----                -
-a---         2015-05-25 오후 9:57             101 testUglify.js
```



```
gulpfile.js  x  testUglify.js  x
1  function uglify(){console.log("저는 어글리해질 운명입니다ㅠㅠ");var o=0;console.log(o)}
```

02. uglify — 난독화하기 — 결과

- 이번엔 Task를 분리해주도록 하겠습니다. 기본 task (default)에서 function()을 두번째 인자로 써줬는데, 다음과 같이 배열명으로 다른 task들의 배열명으로 써줘봅시다. 여기선 uglify하는 기능을 다른 task로 빼봤습니다.

```
gulpfile.js  ×  testUglify.js  ×
1  var gulp = require('gulp');
2  var uglify = require('gulp-uglify');
3
4  gulp.task('default', function() {
5    console.log("Gulp야 안녕 ? ^^; ");
6    gulp.src('js/*.js')
7      .pipe(uglify())
8      .pipe(gulp.dest('build/js'));
9  });
10
```



```

4  // Scripts Task
5  // Uglifies
6  gulp.task('scripts', function(){
7    gulp.src('js/*.js')
8      .pipe(uglify())
9      .pipe(gulp.dest('build/js'));
10 });
11
12 gulp.task('hello', function(){
13   console.log("그냥 인사 한번 해봤어요..")
14 })
15
16 gulp.task('default', ['scripts', 'hello']);
17
```

03. gulp- 다른 Task로 분리

- 다음과 같이 gulp로 했을 때는 지정해준 task들이 진행이 되는 것을 보실 수가 있습니다. (명령어 gulp 로 default 실행됨)

```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\build\js> gulp
[22:15:25] Working directory changed to ~\workspaceHTML\learn_frontend\gulp
[22:15:26] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[22:15:26] Starting 'scripts'...
[22:15:26] Finished 'scripts' after 12 ms
[22:15:26] Starting 'hello'...
그냥 인사 한번 해봤어요..
[22:15:26] Finished 'hello' after 550 µs
[22:15:26] Starting 'default'...
[22:15:26] Finished 'default' after 3.51 µs
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\build\js>
```

- 단순히 이렇게 task를 지정해서 할 수도 있습니다. (명령어 gulp hello)

```
[22:15:26] Finished 'default' after 3.51 µs
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\build\js> gulp hello
[22:15:59] Working directory changed to ~\workspaceHTML\learn_frontend\gulp
[22:16:00] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[22:16:00] Starting 'hello'...
그냥 인사 한번 해봤어요..
[22:16:00] Finished 'hello' after 145 µs
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\build\js> |
```

03. gulp- 다른 Task로 분리 - 결과

- 설치 : `npm install --save-dev gulp-minify-css`
사이트 : <https://github.com/jonathanpollack/gulp-minify-css>
- 용도 : CSS 압축

- 실험할 `css/style.css`파일을 아무렇게 만들어줍시다.
.. 음..근데 좀 쉽네요-_-; 깃헙주소 참고하셔서 이렇게 만들어주시고 `gulp minify-css` 해보시면 되겠습니다~

```

1 body{
2   font-size: 1em;
3   background-color: black;
4 }
5 p{
6   font-color : white;
7 }
8 h1{
9   font-color: red;
10 }

```

```

14 //CSS 최소화
15 var minifyCss = require('gulp-minify-css');
16 gulp.task('minify-css', function() {
17   return gulp.src('css/*.css')
18     .pipe(minifyCss({compatibility: 'ie8'}))
19     .pipe(gulp.dest('dist'));
20 });
21

```

- 결과확인 →

```

1 body{font-size:1em;background-color:#000}p{font-color:#fff}h1{font-color:red}

```

04. CSS minify — CSS 최소화

- 설치: `npm install --save-dev gulp-util .. ^^;`
- 사이트 : <https://github.com/gulpjs/gulp-util>
- 용도 : 로깅용으로 쓰입니다. 자세한 설명은 생략;; 근데 깃헙소스에서는 반환값을 안 받아서 잠시 뭔가..했지만.. 뭐...요것도 간단합니다;;

```
//05. gutil을 통한 로깅
var gutil = require('gulp-util');
gulp.task('loggingtest', function(){
  gutil.log('stuff happened', 'Really it did', gutil.colors.yellow('123')); //주석
  gutil.beep();

  var newPath = gutil.replaceExtension('sample.txt', '.js'); // file.js
  gutil.log("replaceExtension : ", newPath);

  var opt = {
    name: 'todd',
    file: 'js/testUglify.js'
  };
  var tester = gutil.template('test : <%= name %> , file: <%= file %>', opt) // test todd /js/hi.js
  gutil.log("template : ", gutil.colors.red(tester));
});
```

1. 로깅하면서 컬러 지정

2. 확장자 변경하여 반환함

3. 템플릿 지정..

```
[11:50:55] Finished 'loggingtest' after 5.08 ms
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> gulp loggingtest
[11:51:22] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[11:51:22] Starting 'loggingtest'...
[11:51:22] stuff happened Really it did 123
[11:51:22] replaceExtension : sample.js
[11:51:22] template : test : todd , file: js/testUglify.js
[11:51:22] Finished 'loggingtest' after 4.91 ms
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp>
```


- 설치 : `npm install --save-dev gulp-concat`
- 사이트 : <https://github.com/wearefractal/gulp-concat>
- 용도 : **파일간의 연결**(js, css등등)

파일을 만들어주고 깃헙소스대로 한번 써주고.. 테스트 해보겠습니다~

```
//06. concat 으로 js파일 연결
var concat = require('gulp-concat');
gulp.task('concatTest', function() {
  return gulp.src('./concatJs/*.js')
    .pipe(concat('daehanmingukmanse.js'))
    .pipe(gulp.dest('./dist/'));
});
```

```
▼ gulp
  ► build
  ▼ concatJs
    concat01.js
    concat02.js
    concat03.js
```

```
gulpfile.js x concat01.js x concat02.js
1 function concat01(){
2   console.log("내가 대한이다.");
3 }

gulpfile.js x concat02.js x concat01.js
1 function concat02(){
2   console.log("내가 민국이다.");
3 }

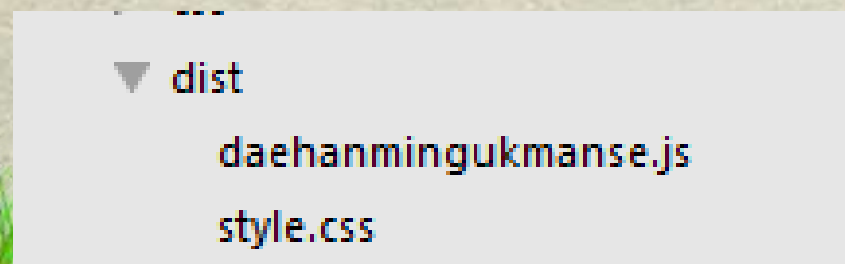
gulpfile.js x concat03.js x concat02.js
1 function concat03(){
2   console.log("내가 만세이다.");
3 }
```


- 흠...뭐 잘 되네요;;
이렇게 파일명을 적어주면서 직접 연결해줘도 됩니다..

```
//return gulp.src(['./concatJs/concat01.js', './concatJs/concat02.js', './concatJs/concat03.js'])  
return gulp.src('./concatJs/*.js')  
  .pipe(concat('daehanmingukmanse.js'))
```

- 실행...

```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> gulp concatTest  
[13:03:04] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js  
[13:03:04] Starting 'concatTest'...  
[13:03:04] Finished 'concatTest' after 22 ms
```



```
gulpfile.js x daehanmingukmanse.js x concat03.js  
1 function concat01(){  
2   console.log("내가 대한이다.");  
3 }  
4 function concat02(){  
5   console.log("내가 민국이다.");  
6 }  
7 function concat03(){  
8   console.log("내가 만세이다.");  
9 }
```

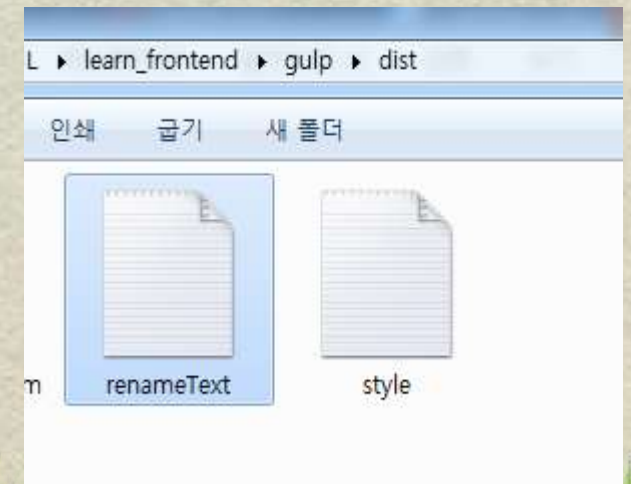
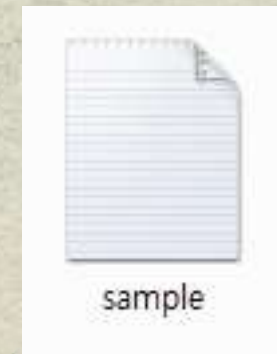
06. gulp-concat

- 가볍게 가보겠습니다. 플러그인이름만 봐도 뭔지 감이 오네요~
- 설치 : `npm install --save-dev gulp-rename`
- 사이트 : <https://github.com/hparra/gulp-rename>
- 용도 : **이름 변경**

- 간단합니다^.^;

```
//07. rename
var rename = require("gulp-rename");
gulp.task('renameTest', function(){
  return gulp.src("sample.txt")
    .pipe(rename("renameText.txt"))
    .pipe(gulp.dest("./dist")); // dist에 복사됨
})
```

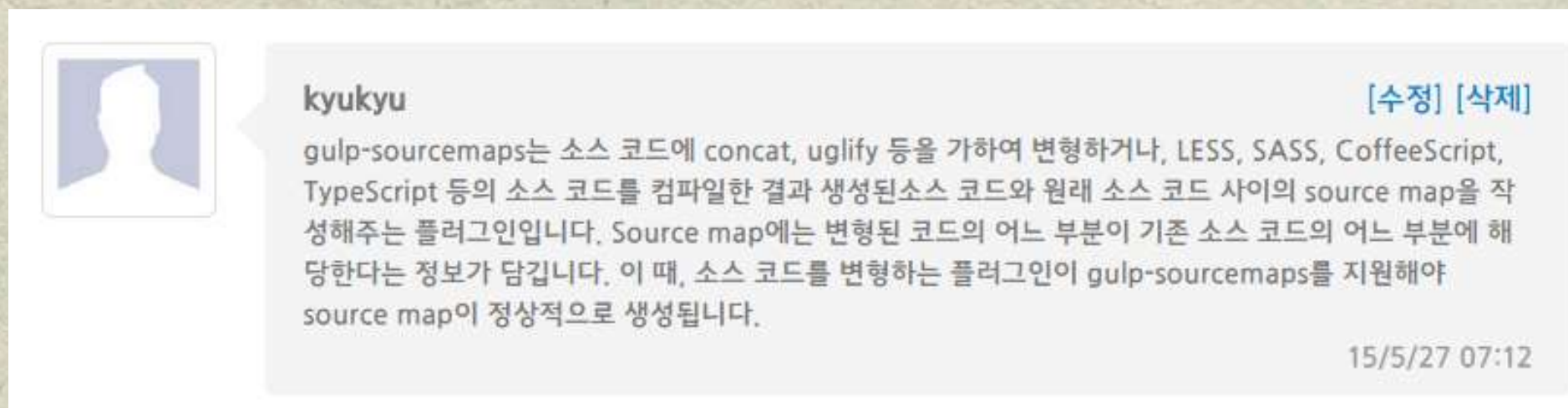
```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> gulp renameTest
[14:25:39] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[14:25:39] Starting 'renameTest'...
[14:25:39] Finished 'renameTest' after 17 ms
```



- 설치 : `npm install --save-dev gulp-sourcemaps`
- 사이트 : <https://github.com/floridoo/gulp-sourcemaps>
- 용도 : 흠 보아하니 우선 눈에 띄는 기능은 여러 작업을 한꺼번에 해주게 하는 것인것 같습니다.
생각해보면, uglify와 concat, rename을 한 번에 할 때는 어떻게 해야 할까요?

- sourcemaps를 통해서 여러 플러그인을 한번에 실행해보겠습니다! 깃헙을 봐도 바로 이해가 안
가므로 코딩을 통해서 알아보겠습니다.

- (수정된 문서입니다!) 역시 개발업계에는 “어떤 것을 아는 가장 빠른 방법은 틀린 것을 말하라”라는 격언대로 개발자분들께서 바로 수정사항에 대한 얘기를 해주셨습니다. 제가 이미 만든 슬라이드는 잘 안 고쳐서;; 댓글을 바로 사진으로 붙여넣겠습니다. 슬라이드상으로나마 다시 한번 감사의 말씀을 드립니다^^; 아웃사이드님의 소스맵에 관한 링크는 이쪽으로;; <http://blog.outsider.ne.kr/916>

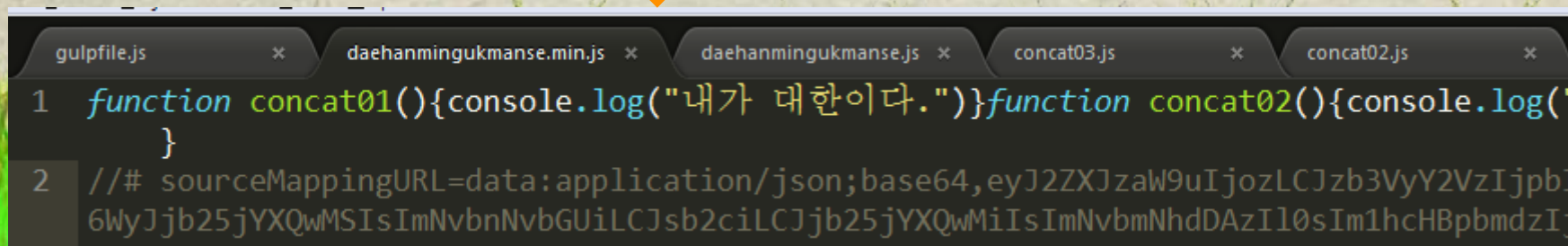


08. gulp-sourcemaps

- **흐음.. git 설명을 볼때는 뭔가 했는데, 조합을 하니.. .자..잘 됩니다 😊 (간단데스.. 괜히..이 문서를 만드는건가;;)**
- **모아서, 난독화시키고, 이름 변경까지 땡!**
init과 write사이에 플러그인 명령들을 땡!땡~ 써주면 됩니다.

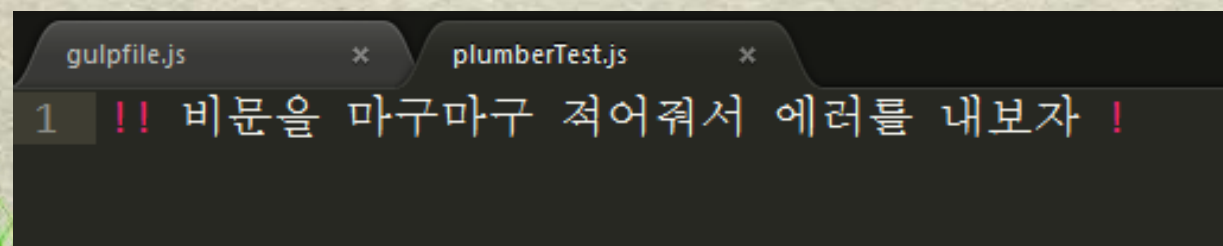
```
//08. sourcemaps로 source map을 지원하는 여러 작업 한번에 처리
var sourcemaps = require('gulp-sourcemaps');
gulp.task('sourcemapsTest', function() {
  gulp.src('./concatJs/*.js')
    .pipe(sourcemaps.init())
    .pipe(concat('daehanmingukmanseVersion2.js'))
    .pipe(uglify())
    .pipe(rename('daehanmingukmanse.min.js'))
    .pipe(sourcemaps.write())
    .pipe(gulp.dest('dist'));
});
```

이 사이에 작업들이 들어갑니다^^



- 설치 : `npm install --save-dev gulp-plumber`
- 사이트 : <https://github.com/floatdrop/gulp-plumber>
- 용도 : **에러핸들링**

- 각종 작업을 하다보면 에러가 나는 경우가 있습니다. 하지만 에러메시지가 불친절한 경우가 있는데요. 친절한 에러메시지는 디버깅을 도와주기 때문에, 꼭 필요합니다. 플럼버를 통해 에러메시지를 친절하게 해보겠습니다.
- 우선 에러메시지를 유발할 `test/plumberTest.js`파일을 다음과 같이 만들어보겠습니다.



```
gulpfile.js x plumberTest.js x
1 !! 비문을 마구마구 적어줘서 에러를 내보자 !
```

09. gulp-plumber 에러핸들링

- 소스를 다음과 같이 적어줘보겠습니다. 우선은 플럼버의 에러핸들링 구문을 안 적어주고 실행해보면 다음과 같은 결과가 나옵니다.
- 어디서 에러가 났는지 찾아보기가 힘듭니다.

```
// 09. Plumber 를 이용한 에러핸들링
gulp.task('plumberTest', function() {
  return gulp.src('./test/plumberTest.js')
    .pipe(plugings.plumber({
      errorHandler: onError
    }))
    .pipe(plugings.uglify())
    .pipe(gulp.dest('build/js'));
});
// plumber 에러 핸들러 함수를 외부에 정의해놓겠습니다.
var onError = function(err) {
  console.log(err);
}
```

```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> gulp plumberTest
[15:15:00] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[15:15:00] Starting 'plumberTest'...

events.js:85
    throw er; // Unhandled 'error' event
    ^
Error
    at new JS_Parse_Error (C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\node_modules\acorn\dist\acorn.js:926:17)
    at js_error (C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\node_modules\acorn\dist\acorn.js:951:17)
    at croak (C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\node_modules\acorn\dist\acorn.js:961:17)
    at token_error (C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\node_modules\acorn\dist\acorn.js:971:17)
    at unexpected (C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\node_modules\acorn\dist\acorn.js:981:17)
    at semicolon (C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\node_modules\acorn\dist\acorn.js:991:17)
    at simple_statement (C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\node_modules\acorn\dist\acorn.js:1001:17)
    at C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\node_modules\acorn\dist\acorn.js:1011:17
    at C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\node_modules\acorn\dist\acorn.js:1021:17
    at C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\node_modules\acorn\dist\acorn.js:1031:17
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> |
```

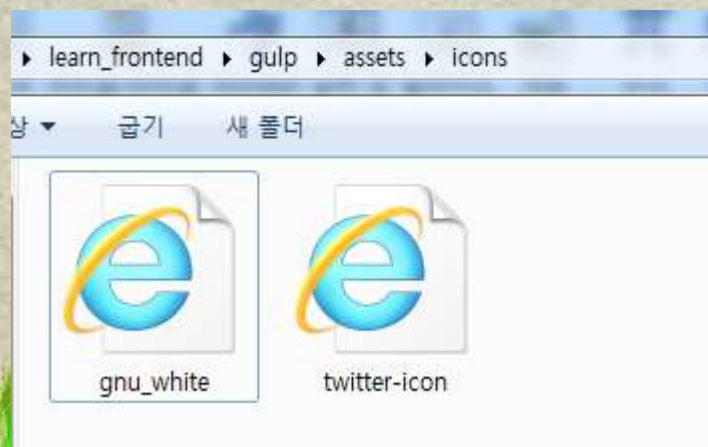

- 이번엔 plumber 주석을 해제해주고 한번 살펴보겠습니다...
- 다음과 같이 친절한 에러메시지가 나옵니다.^^;
- 애용해야겠습니다.. 플럼버!

```
// 09. Plumber 를 이용한 에러 핸들링
var plumber = require('gulp-plumber');
gulp.task('plumberTest', function() {
  return gulp.src('./js/plumberTest.js')
    .pipe(plumber({
      errorHandler: onError
    }))
    .pipe(uglify())
    .pipe(gulp.dest('build/js'));
});
// plumber 에러 핸들러 함수를 외부에 정의해놓겠습니다
var onError = function(err) {
  console.log(err);
}
```

```
[21:50:33] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[21:50:33] Starting 'plumberTest'...
{ [Error: C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\test\plumberTest.js: Unexpected token: name (마구마구)]
  message: 'C:\\Users\\arahansa\\workspaceHTML\\learn_frontend\\gulp\\test\\plumberTest.js: Unexpected token: name (마-
  fileName: 'C:\\Users\\arahansa\\workspaceHTML\\learn_frontend\\gulp\\test\\plumberTest.js',
  lineNumber: 1,
  stack: 'Error\\n    at new JS_Parse_Error (C:\\Users\\arahansa\\workspaceHTML\\learn_frontend\\gulp\\node_modules\\gul
96:18)\\n    at js_error (C:\\Users\\arahansa\\workspaceHTML\\learn_frontend\\gulp\\node_modules\\gulp-uglify\\node_modu
k (C:\\Users\\arahansa\\workspaceHTML\\learn_frontend\\gulp\\node_modules\\gulp-uglify\\node_modules\\uglify-js\\lib\\p
ahansa\\workspaceHTML\\learn_frontend\\gulp\\node_modules\\gulp-uglify\\node_modules\\uglify-js\\lib\\parse.js:683:9)\\n
TML\\learn_frontend\\gulp\\node_modules\\gulp-uglify\\node_modules\\uglify-js\\lib\\parse.js:689:9)\\n    at semicolon (
\\gulp\\node_modules\\gulp-uglify\\node_modules\\uglify-js\\lib\\parse.js:709:43)\\n    at simple_statement (C:\\Users\\
e_modules\\gulp-uglify\\node_modules\\uglify-js\\lib\\parse.js:889:73)\\n    at C:\\Users\\arahansa\\workspaceHTML\\lear
modules\\uglify-js\\lib\\parse.js:757:20\\n    at C:\\Users\\arahansa\\workspaceHTML\\learn_frontend\\gulp\\node_modules
.js:722:24\\n    at C:\\Users\\arahansa\\workspaceHTML\\learn_frontend\\gulp\\node_modules\\gulp-uglify\\node_modules\\u
  showStack: false,
  showProperties: true,
  plugin: 'gulp-uglify' }
[21:50:34] Finished 'plumberTest' after 170 ms
```


- 설치 : `npm install --save-dev gulp-iconfont`
- 사이트 : <https://github.com/nfroidure/gulp-iconfont>
- 용도 : **svg파일을 font로 만들어줍니다.**

- 음.. 처음봐서는 감이 잘 오진 않네요. 코딩하고 돌려보면서보겠습니다. `assets/icon` 폴더에 다음의 두 **svg**파일을 받아주겠습니다.



10. **gulp-iconfont** svg를 폰트로

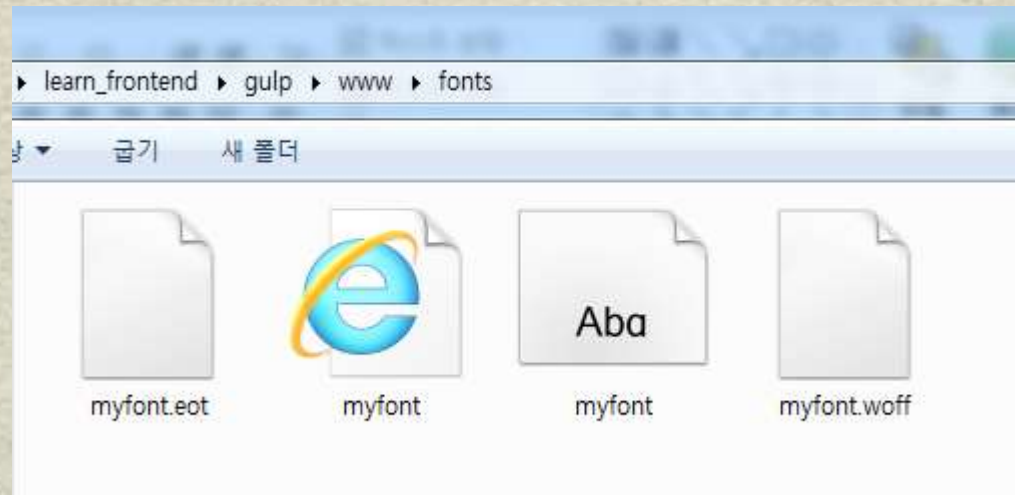
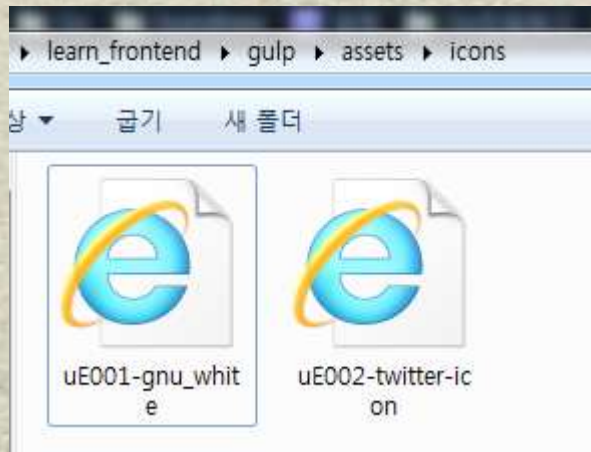
- 다음과 같은 코드로 svg이미지를 **www/fonts**폴더에 font로 변환시켜주겠습니다.

```
// 10. Iconfont를 통한 svg파일 웹폰트 변환
var iconfont = require('gulp-iconfont');
gulp.task('Iconfont', function(){
  gulp.src(['assets/icons/*.svg'])
    .pipe(iconfont({
      fontName: 'myfont', // required
      appendCodepoints: true // recommended option
      //fontHeight : 1000
      // normalize:true
    }))
    .on('codepoints', function(codepoints, options) {
      // CSS templating, e.g.
      console.log(codepoints, options);
    })
    .pipe(gulp.dest('www/fonts/'));
});
```

```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> gulp Iconfont
[16:56:34] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[16:56:34] Starting 'Iconfont'...
[16:56:34] Finished 'Iconfont' after 11 ms
[ { name: 'gnu_white', codepoint: 57345 },
  { name: 'twitter-icon', codepoint: 57346 } ] { fontName: 'myfont',
  appendCodepoints: true,
  ignoreExt: false,
  log: [Function],
  error: [Function],
  fixedWidth: false,
  descent: 0,
  round: 1000000000000000 }
[16:56:34] gulp-svgicons2svgfont: The provided icons does not have the same
m.
[16:56:34] gulp-svgicons2svgfont: Font created
[16:56:34] Saved codepoint: uE001 for gnu_white.svg
[16:56:34] Saved codepoint: uE002 for twitter-icon.svg
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp>
```

10. gulp-iconfont

- 결과 : 다음과 같이 assets/icons 폴더의 svg이미지들은 파일명이 이렇게 변했고~ gulp 설정에서 명시해준 대로 www/fonts에서는 다음과 같은 파일들이 생겼습니다.



- 이것을 이제 어떻게 사용해볼까요? 다음의 사이트를 참조해서, 한번 사용을 해보겠습니다.

<http://gomakethings.com/icon-fonts/>

10. gulp-iconfont svg를 폰트로

다음과 같이 간단히
html파일을 만들어주어서
적용해보겠습니다.


파일명 icontest.html

You'll notice that we're using the CSS3 `:before` selector. This inserts the specified content `before` whatever HTML object the class is applied to. The `content` property is used to define the unicode character.

before를 사용하는 이유는
위과 같습니다.

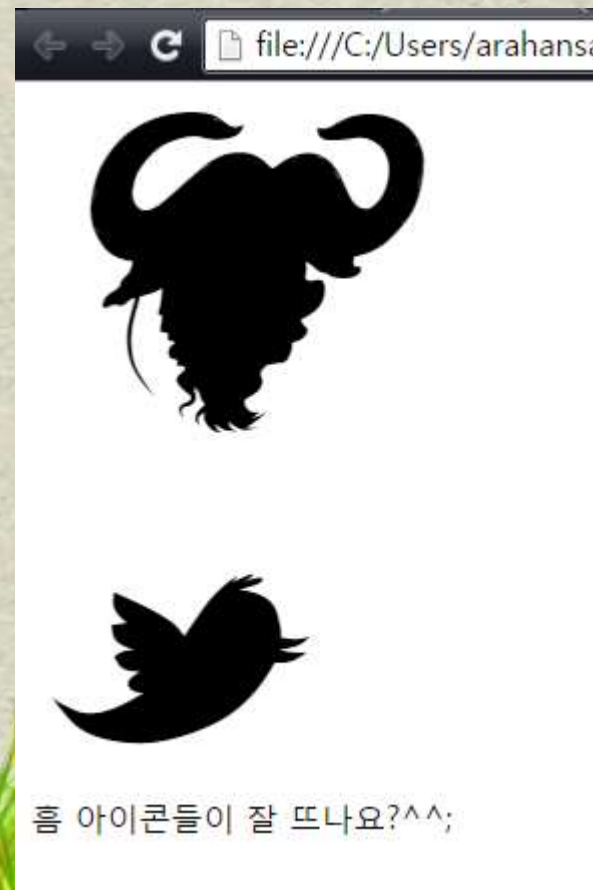
스타일에서 클래스 안에 들어갈
content에는
이전 슬라이드에서 svg파일명이
바뀌었는데 거기 나온 번호를
넣어줍니다.

```
gulpfile.js  icontest.html  plumberTest.js
1  <html>
2    <head>
3      <meta charset="utf-8">
4      <title>gulp 테스트</title>
5      <style>
6        /* Icons */
7        @font-face {
8          font-family: 'myfont';
9          src:url('www/fonts/myfont.eot');
10         src:url('www/fonts/myfont.eot') format('embedded-opentype'),
11           url('www/fonts/myfont.ttf') format('truetype'),
12           url('www/fonts/myfont.woff') format('woff'),
13           url('www/fonts/myfont.svg') format('svg');
14         font-weight: normal;
15         font-style: normal;
16       }
17       .icon-twitter:before {
18         font-size: 10em;
19         font-family: 'myfont';
20         content: "\e001";
21       }
22       .icon-gnu:before {
23         font-size: 10em;
24         font-family: 'myfont';
25         content: "\e002";
26       }
27     </style>
28   </head>
29   <body>
30     <i class="icon-twitter"></i><br>
31     <i class="icon-gnu"></i><br>
32     흙
33   </body>
34 </html>
```



10. gulp-iconfont svg를 폰트로

- <i class="클래스명"> 으로 아이콘들이 잘 불러와진 것을 알 수 있습니다. 신기하네요~
(아 근데 원래 흑백인가요-_-?) 흠.. 아..아무튼..뭐 이런거구나 하고 패쓰해봅니다;; (당장 안 쓰므로...무책임;;)



10. gulp-iconfont 결과물

- iconfont 깃헙사이트에는
css도 자동으로 생성해주는 것
같은 옵션이 뜬니다만,
- 이상하게 저는 자꾸 실행하다가
lodash에서 에러가 나는군요
- 당장 필요한 건 아니니 넘어갑시다만,
혹시 이렇게 하시는 분 계시면
알려주시면 감사하겠습니다^^

Make your CSS

To use this font in your CSS, you could add a mixin like in this [real world](#) automatically with [gulp-iconfont-css](#).

It's also easy to make a CSS template by yourself. Like [this example](#), [gulp template](#).

```
var gulp = require('gulp');
var iconfont = require('gulp-iconfont');
var consolidate = require('gulp-consolidate');

gulp.task('iconfont', function(){
  gulp.src(['assets/icons/*.svg'])
    .pipe(iconfont({ fontName: 'myfont' }))
    .on('codepoints', function(codepoints, options) {
      gulp.src('templates/myfont.css')
        .pipe(consolidate('lodash', {
          glyphs: codepoints,
          fontName: 'myfont',
          fontPath: '../fonts/',
          className: 's'
        }))
        .pipe(gulp.dest('www/css/'));
    })
    .pipe(gulp.dest('www/fonts/'));
});
```


- 설치 : `npm install --save-dev gulp-autoprefixer`
- 사이트 :
<https://www.npmjs.com/package/gulp-autoprefixer>
<https://github.com/postcss/autoprefixer>
- 용도 : **벤더프리픽스 자동생성**? 일단 npmjs에서 다음과 같은 주소를 받아서 한번 돌려보겠습니다. 옵션이 좀 있는데요... browsers는 브라우저 어디까지 프리픽스를 생성할 것인지 설정입니다.

```
5
6 // 11. autoPrefixer
7 var autoprefixer = require('gulp-autoprefixer');
8 gulp.task('autoPrefixer', function () {
9     return gulp.src('css/app.css')
10         .pipe(autoprefixer({
11             browsers: ['last 2 versions'],
12             cascade: false
13         }))
14         .pipe(gulp.dest('dist'));
15 });
16
```

Queries

You can specify the versions by queries (case insensitive):

- last 2 versions : the last 2 versions for each major browser.
- last 2 Chrome versions : the last 2 versions of Chrome browser.
- > 5% : versions selected by global usage statistics.
- > 5% in US : uses USA usage statistics. It accepts [two-letter country code](#).
- Firefox > 20 : versions of Firefox newer than 20.
- Firefox >= 20 : versions of Firefox newer than or equal to 20.
- Firefox < 20 : versions of Firefox less than 20.
- Firefox <= 20 : versions of Firefox less than or equal to 20.
- Firefox ESR : the latest [Firefox ESR] version.
- ios 7 : the iOS browser version 7 directly.

11. gulp-autoprefixer

- 참고로 autoprefixer 역시 sourcemaps 를 지원해서 다른 플러그인들과 같이 사용할 수 있습니다.
- css/app.css를 다음과 같이 만들어주시고 설정한 대로 gulp autoPrefixer를 실행시켜보면...다음과 같이 dist 폴더에 app.css가 생깁니다.

```
gulpfile.js x app.css — css x app.css —
1 a {
2   display: flex;
3   -webkit-border-radius: 5px;
4   border-radius: 5px;
5 }
6
```



```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> gulp autoPrefixer
[20:32:33] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[20:32:33] Starting 'autoPrefixer'...
[20:32:33] Finished 'autoPrefixer' after 50 ms
```



```
1 a {
2   display: -webkit-flex;
3   display: -ms-flexbox;
4   display: flex;
5   border-radius: 5px;
6 }
7
```

11. gulp-autoprefixer 결과

- 설치 : `npm install --save-dev gulp-load-plugins`
- 사이트 : <https://www.npmjs.com/package/gulp-load-plugins>
- 용도 : 플러그인 관리. 불러오기 설정.

- 지금까지 많은 플러그인들을 직접 `require`하셔서 받아오셨는데요. 이것도 프로그래머에게는 무척 귀찮은 일입니다.-_- 이것도 쉽게 만들어보겠습니다. 깃헙을 참고해서 다음과 같이 만들어주겠습니다.
- 다음과 같이 적어주시면 오른쪽 사진처럼 다른 플러그인들을 일일이 `require`해주는 작업을 안해줘도 됩니다. (단 뒷장의 json을 미리 정의해야합니다)

```
// 12. gulp-load-plugins를 통한 플러그인 관리
var gulpLoadPlugins = require('gulp-load-plugins');
var plugins = gulpLoadPlugins(
  //옵션들 여기다가 넣어줌
);
```

```
//05. gutil을 통한 로깅
var gutil = require('gulp-util');
```

12. gulp-load-plugins

- 다만, 좌측하단 사진과 같이 package.json에 필요한 플러그인파일들을 적어주셔야 합니다. (npm install)은 미리 해주신 상태..
- 그리고 기존의 플러그인들을 쓰실때는 plugins 이후로 기존플러그인 이름을 붙여주시면 됩니다. gulp-autoPrefixer 의 경우에는 plugins.autoprefixer가 되는 것입니다. 그리고 기존의 task들을 돌려보시면 아주 잘 동작한다는 것을 느끼실 수 있습니다 ^^

```
gulpfile.js  x  package.json  x
1  {
2    "dependencies": {
3      "gulp-autoprefixer": "*",
4      "gulp-iconfont": "*",
5      "gulp-plumber": "*",
6      "gulp-sourcemaps": "*",
7      "gulp-rename": "*",
8      "gulp-concat": "*",
9      "gulp-util": "*",
10     "gulp-uglify": "*",
11     "gulp-minify-css": "*"
12   }
13 }
14
```

```
// 12. gulp-load-plugins를 통한 플러그인 관리
var gulpLoadPlugins = require('gulp-load-plugins');
var plugins = gulpLoadPlugins(
  // 옵션들 여기다가 넣어줌
);
```

↓

```
9 // 11. autoPrefixer
10 gulp.task('autoPrefixer', function () {
11   return gulp.src('css/app.css')
12     .pipe(plugins.autoprefixer({
13       browsers: ['last 2 versions'],
14       cascade: false
15     }));
16 }
```

12. gulp-load-plugins 결과

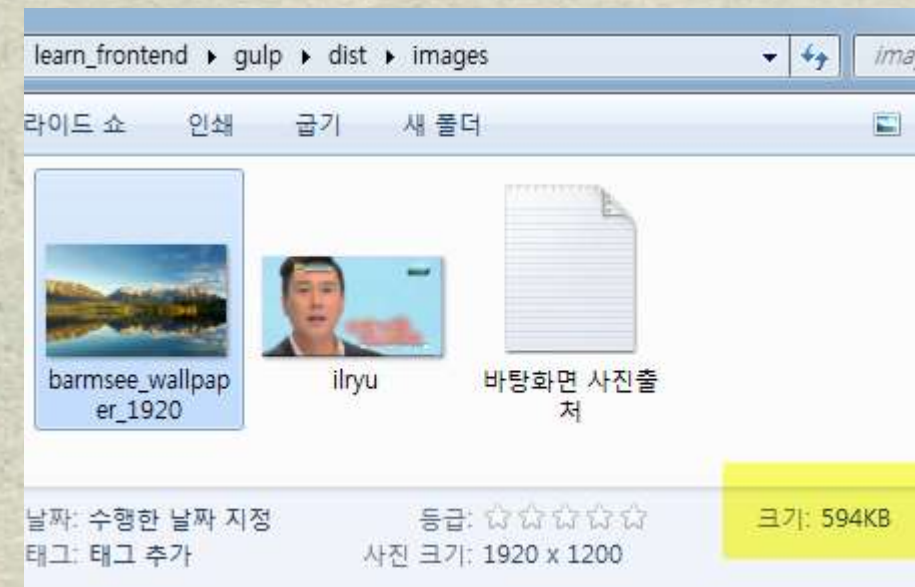
- 설치 : `npm install --save-dev gulp-imagemin`
- 사이트 : <https://github.com/sindresorhus/gulp-imagemin>
- 용도 : 이미지를 압축한다!

```
4 // 13. 이미지 압축
5 // 그냥 자동 플러그인 안 써주고 명시적으로 적어주겠습니다 ^^;
6 var imagemin = require('gulp-imagemin');
7 var pngquant = require('imagemin-pngquant');
8 gulp.task('imgmin', function () {
9     return gulp.src('src/images/*')
10         .pipe(imagemin({
11             progressive: true,
12             svgoPlugins: [{removeViewBox: false}],
13             use: [pngquant()]
14         }))
15         .pipe(gulp.dest('dist/images'));
16 });
17
```

13. `gulp-imagemin` 이미지 압축

- 압축..뭐.. 간단히 됩니다. 옵션중에는 무손실전환? 최적화 레벨등 다양한 옵션이 있습니다~

```
[23:51:45] Finished 'imgmin' after 89 ms
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\build\js> gulp imgmin
[23:59:41] Working directory changed to ~\workspaceHTML\learn_frontend\gulp
[23:59:45] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[23:59:45] Starting 'imgmin'...
[23:59:45] gulp-imagemin: Minified 2 images (saved 41.67 kB - 6.3%)
[23:59:45] Finished 'imgmin' after 299 ms
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\build\js>
```



옵션의 예 :

progressive (jpg)

Type: boolean

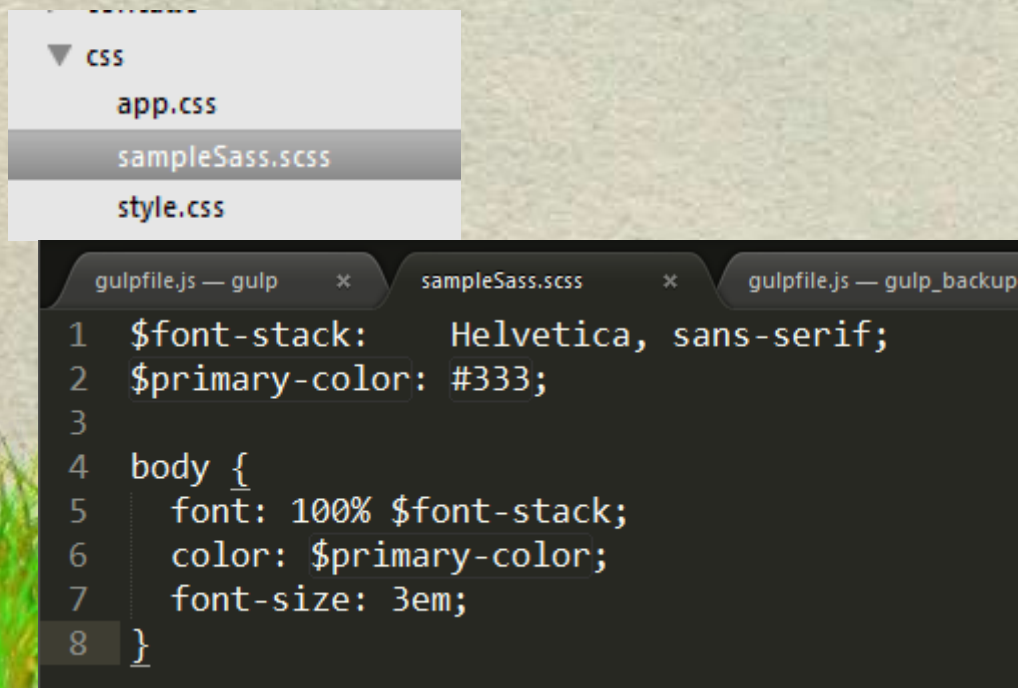
Default: false

Lossless conversion to progressive.

13. gulp-imagemin 이미지 압축 결과

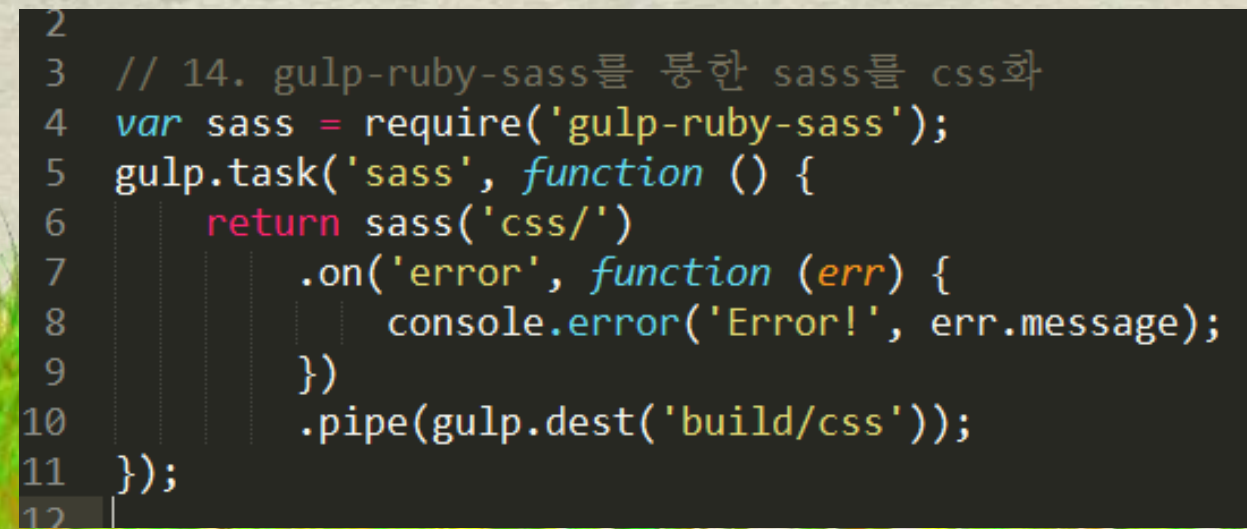
- 설치 : `npm install --save-dev gulp-ruby-sass`
- 사이트 : <https://github.com/sindresorhus/gulp-ruby-sass>
- 용도 : **sass 컴파일링**

- 흠. sass .. 요건 css를 좀 더 편하게 적기 위해서.. 나온 걸로 less, sass, stylus 뭐 이런 것이 있는 것같지만.. 아무튼 여기선 sass로 해봅시다; css/sampleSass.scss파일을 만들어주고 gulffile을 다음과 같이 설정해보겠습니다.



```
▼ css
  app.css
  sampleSass.scss
  style.css

gulpfile.js — gulp  x  sampleSass.scss  x  gulpfile.js — gulp_backup
1 $font-stack: Helvetica, sans-serif;
2 $primary-color: #333;
3
4 body {
5   font: 100% $font-stack;
6   color: $primary-color;
7   font-size: 3em;
8 }
```



```
2
3 // 14. gulp-ruby-sass를 통한 sass를 css화
4 var sass = require('gulp-ruby-sass');
5 gulp.task('sass', function () {
6   return sass('css/')
7     .on('error', function (err) {
8       console.error('Error!', err.message);
9     })
10    .pipe(gulp.dest('build/css'));
11 });
12
```

14. **gulp-ruby-sass** sass를 css화

- sass를 돌리면 이렇게 됩니다. 아. 간단하군요...;;
갑자기 괜히 이걸 적나 하는 생각이... 그럼 다음번엔
좀 더 신기한 걸로 가볼까요?

```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\build\js> gulp sass
[01:19:36] Working directory changed to ~\workspaceHTML\learn_frontend\gulp
[01:19:39] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[01:19:39] Starting 'sass'...
[01:19:40] directory ./
[01:19:40] write ./sampleSass.css
[01:19:40] Finished 'sass' after 499 ms
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\build\js> |
```

```
gulpfile.js — gulp × sampleSass.scss × gulpfile.js — gulp_backup
1 $font-stack: Helvetica, sans-serif;
2 $primary-color: #333;
3
4 body {
5   font: 100% $font-stack;
6   color: $primary-color;
7   font-size: 3em;
8 }
```



```
gulpfile.js × sampleSass.scss × sampleSass.css ×
1 body {
2   font: 100% Helvetica, sans-serif;
3   color: #333;
4   font-size: 3em; }
5
```


- 이번엔 외부 플러그인이 아닌 Gulp에 내장된 watch라는 기능을 이용해보겠습니다. **파일 변경을 실시간으로 감지**하면서 바뀔때마다 정해진 작업을 해주는 기능입니다. 이렇게 써주시고 `gulp watch` 를 해보겠습니다.

```
//15. watch를 통한 실시간 변경
gulp.task('watch', function(){
  //css폴더의 scss파일들이 변경되면 sass태스크 실행
  gulp.watch('css/*.scss', ['sass']);
});
```

- `watch`를 걸어주면 다음과 같이 대기모드가 됩니다.

```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\build\js> gulp watch
[01:32:37] Working directory changed to ~\workspaceHTML\learn_frontend\gulp
[01:32:40] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[01:32:40] Starting 'watch'...
[01:32:40] Finished 'watch' after 8.24 ms
```

15. **watch** 를 통한 실시간 변경

- scss파일에서 색깔을 다음과 같이 red로 변경해줘볼까요?
다음과 같이 변경감지가 되면서 다시 쓰기를 해줍니다.

```
gulpfile.js — gulp × sampleSass.scss × sampleSass.css
1 $font-stack: Helvetica, sans-serif;
2 $primary-color: red;
3
4 body {
5   font: 100% $font-stack;
6   color: $primary-color;
7   font-size: 3em;
8 }
```



```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\build\js> gulp watch
[01:32:37] Working directory changed to ~\workspaceHTML\learn_frontend\gulp
[01:32:40] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[01:32:40] Starting 'watch'...
[01:32:40] Finished 'watch' after 8.24 ms
[01:34:26] Starting 'sass'...
[01:34:26] directory ./
[01:34:26] write ./sampleSass.css
[01:34:26] Finished 'sass' after 362 ms
```



- watch 또한 sourcemaps으로 연결을 하여 사용할 수 있습니다.
실제로 작업할 때는 watch를 켜두고서 작업을 하면 되겠죠~?

```
gulpfile.js — gulp × sampleSass.scss × sampleSass.css ×
1 font: 100% Helvetica, sans-serif;
2 color: red;
3 font-size: 3em;
```

15. watch 를 통한 실시간 변경

- 설치 : `npm install browser-sync gulp --save-dev`
 - 사이트 : <http://www.browsersync.io/docs/gulp/>
 - 용도 : 웹서버를 작동하면서, 자동 Reload해주고 다양한 기기(컴퓨터, 태블릿, 모바일) 에서 화면을 보게 해준다 (화면 위치추적까지해줌)
-
- https://www.youtube.com/watch?v=Rv5f_v5gqw4
 - 여기를 한번 봐주시죠~ 백마디 말보다 한 번 보는 게 나을 것 같습니다.
 - 장점 : **백엔드 서버 없이** 프론트엔드 화면을 자체적으로 바로 볼 수 있으며(클라이언트 렌더링), **설정을 바꿀때마다 자동으로 Reload**됩니다. 위치추적으로 **다양한 디바이스**를 켜고서 화면을 따라가기가 좋습니다.
 - 설정부터 들어가겠습니다. 공홈의 소스를 참고해서 다음과 같이 수정해줍니다.

15. BrowerSync - 웹서버 작동과 다양한 디바이스 적용.

- 다음과 같이 공홈에서 참조하여 써주겠습니다.

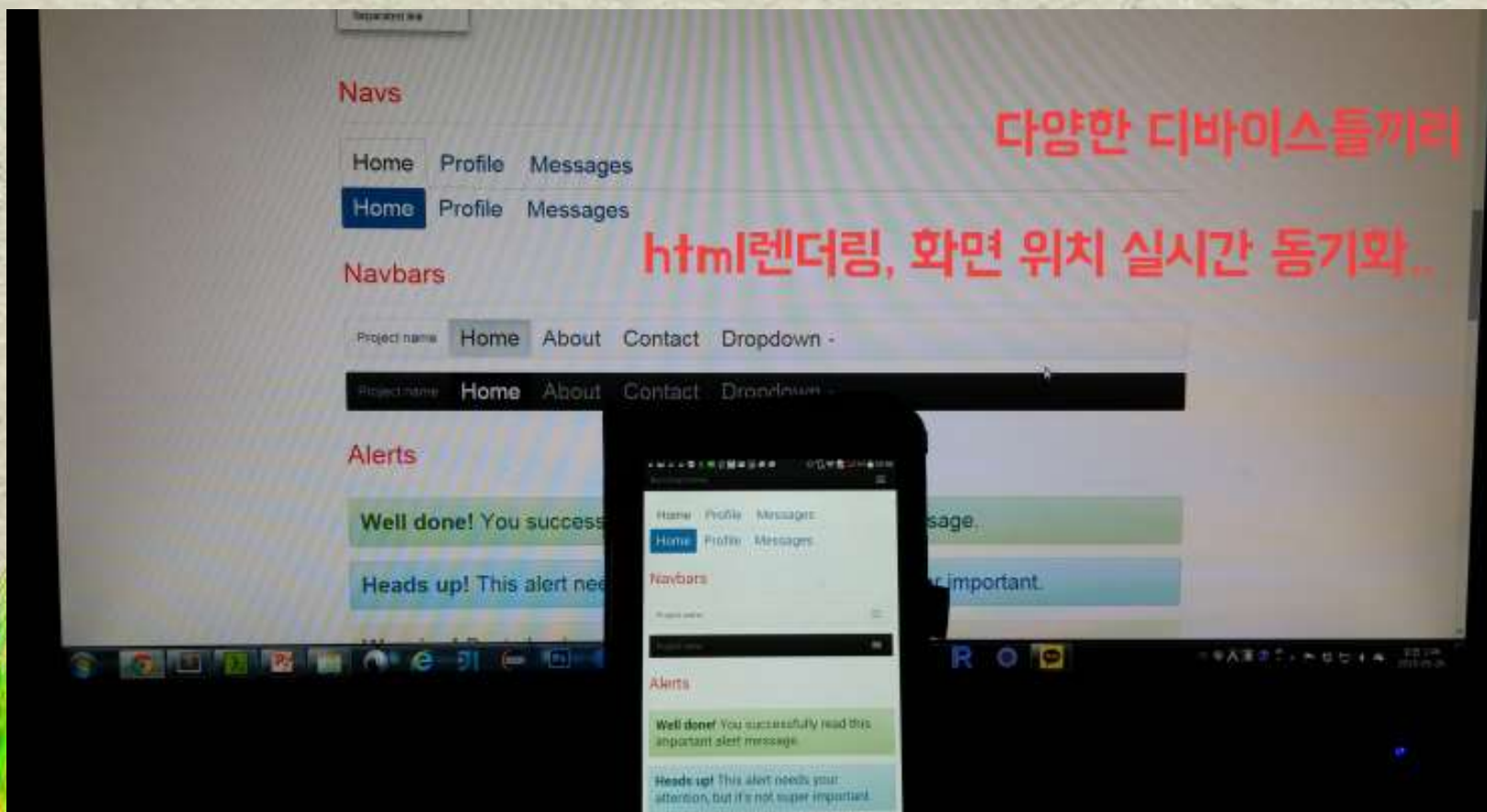
```
2
3 // 16. brower-sync
4 var browserSync = require('browser-sync').create();
5 var reload      = browserSync.reload;
6 // Static server
7 gulp.task('browser-sync', function() {
8     browserSync.init({
9         server: {
10             baseDir: "./"
11         }
12     });
13     gulp.watch("css/*.scss", ['sassForbs']);
14     gulp.watch("./*.html").on('change', reload);
15 });
16
```

```
// 공홈에서 퍼와서 약간 수정
gulp.task('sassForbs', function () {
    return sass('css', {sourcemap: true})
        .on('error', function (err) {
            console.error('Error!', err.message);
        })
        .pipe(plugings.sourcemaps.write('./', {
            includeContent: false,
            sourceRoot: 'css'
        }))
        .pipe(gulp.dest('build/css'))
        // match(css)에 해당하는 것 변환시 stream으로 sync 맞춰줌
        .pipe(browserSync.stream({match: '**/*.css'}));
});
```

- 소스가 약간 복잡해서 설명을 드리자면 왼쪽사진에서는 브라우저싱크 맞추는데, html같은경우는 알아서 watch를 하면서 reload변수로 change를 합니다. scss의 변경감지의 경우 오른쪽 task를 실행하면서 css에 써주게 되고, 브라우저싱크에 매칭시켜서 브라우저를 다시불러오게 해줍니다. (흠.. 공홈소스 그대로 퍼왔는데, 왠지 옵션이 거슬리네요;; 암튼 넘어갑시다;)

- 이제 한번 돌려볼까요? 다음과 같이 서버가 켜지면서 실시간으로 다양한 기기에서 화면이 공유가 됩니다.

```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> gulp browser-sync
[01:42:38] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[01:42:38] Starting 'browser-sync'...
[01:42:38] Finished 'browser-sync' after 25 ms
[BS] Access URLs:
  -----
    Local: http://localhost:3000
    External: http://192.168.0.4:3000
  -----
    UI: http://localhost:3001
    UI External: http://192.168.0.4:3001
  -----
[BS] Serving files from: ./
```



- scss파일을 변경하면 css파일이 변경되면서 실시간으로 화면도 바뀝니다.

```
gulpfile.js — gulp × sampleSass.scss × sampleSass.css ×
1 $font-stack: Helvetica, sans-serif;
2 $primary-color: black;
3
4 body {
5   font: 100% $font-stack;
6   color: $primary-color;
7   font-size: 3em;
8
9 }
```



- 이제 프론트엔드 작업들은 백엔드작업과는 별개로 자체적인 서버를 구동시키면서 이와 같은 작업흐름을 가져갈 수 있습니다...흠..일추 계획한 분량보다 조금 많게 끝나가는데요...

- 일단 10개 정도만 하려했는데.. 이왕 한 김에 좀 더 가봅니다.
테스트, 유닛테스트, 품질도구같은 것들까지 보고 갑시다~
(설문조사 결과가 오덕후가 나와서...-_-;;흠....이상하네요... 이럴리가 없..오 벗어나는데...;))

나는 덕후일까?

초오타쿠

나는 덕후일까?

취향입니다. 존중해 주시죠?

초 오타쿠

당신의 삶은 이미 덕질로 물들었습니다. 가끔은 주위를 둘러보며 가족도 챙기고 친구들도 만나야 합니다. 어이 당신! 현실 세계를 잊지 말아달라구!

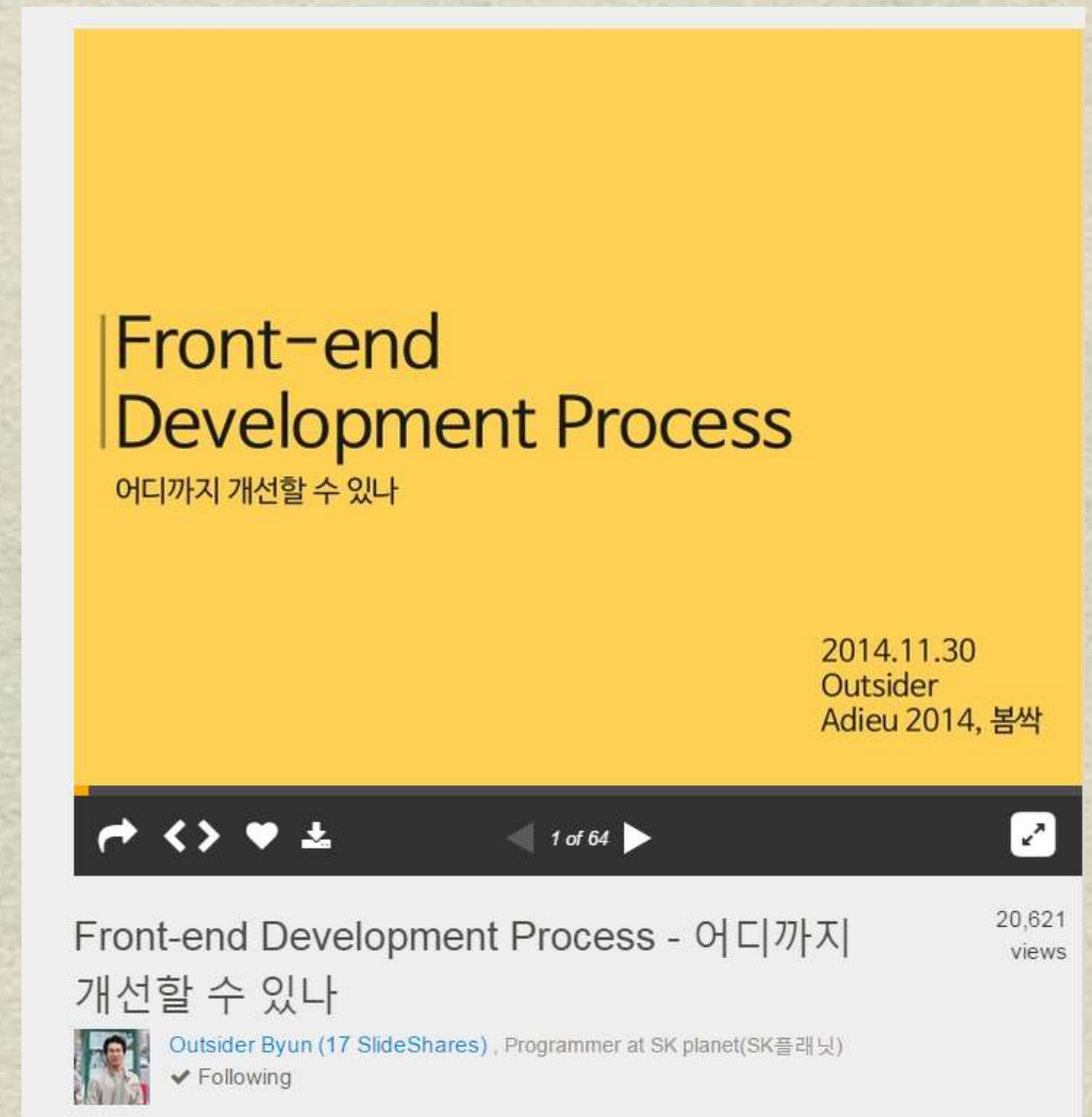
잠시 쉬어가며.. 내..내가 왜 덕후인거지-_-...흠

- Outsider님의 봄씩세미나 슬라이드에서 테스트에 대한 것까지 다루시는 슬라이드를 보고,

이 슬라이드에서도 이왕 하는거 테스트까지 해보자-_-라고 생각해보게 되었습니다.

- 아직 안 보신분은 참조를..

<http://www.slideshare.net/rockdoli/adieu-springsprout2014>



아..미리 적고 가자면 테스트 부분 영감을 받은
곳은...

설치 : npm install gulp-jshint --save-dev

사이트 : <https://github.com/spalger/gulp-jshint>

용도 : 자바스크립트 코드 품질 관리

뭐..지금까지 흐름은 비슷하니, 일단 jshint만 보겠습니다. csshint, htmlhint 여러가지 힌트가 많네요^^; 알아서 보..보세요(무책임)

```
// 17. jshint
var jshint = require('gulp-jshint');
gulp.task('lint', function() {
  return gulp.src('./test/jshinttest.js')
    .pipe(jshint())
    .pipe(jshint.reporter('YOUR_REPORTER_HERE'));
});
```

```
[02:30:42] Please check the documentation for proper gulpfile formatting
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> gulp lint
[02:30:43] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[02:30:43] Starting 'lint'...
C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\test\jshinttest.js: line 3, col 19, Missing semicolon.

1 error
[02:30:43] Finished 'lint' after 39 ms
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> |
```

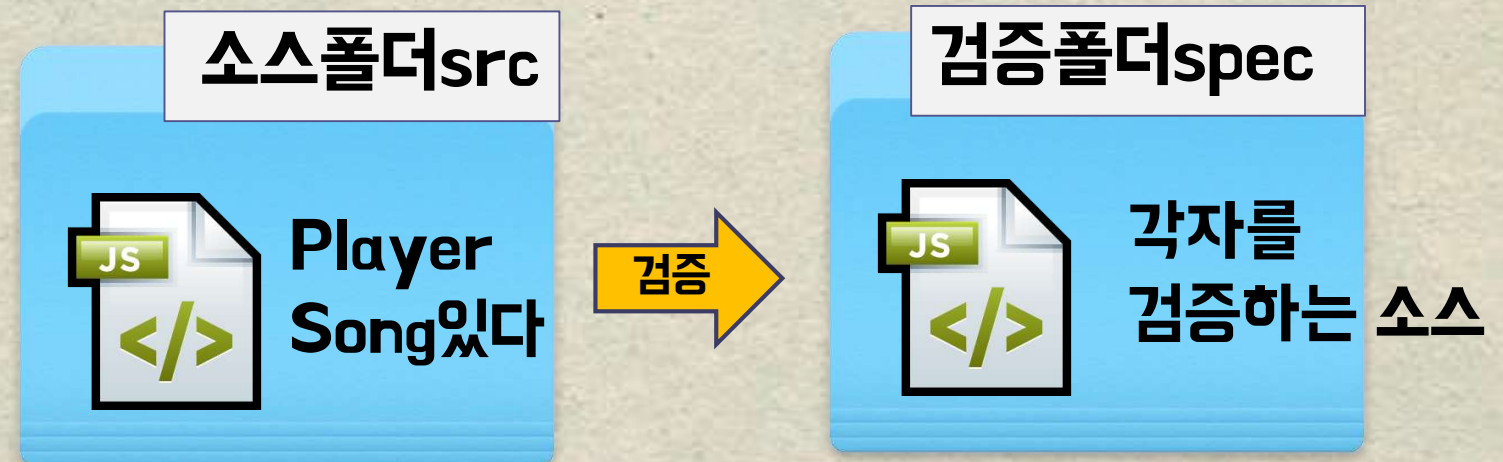
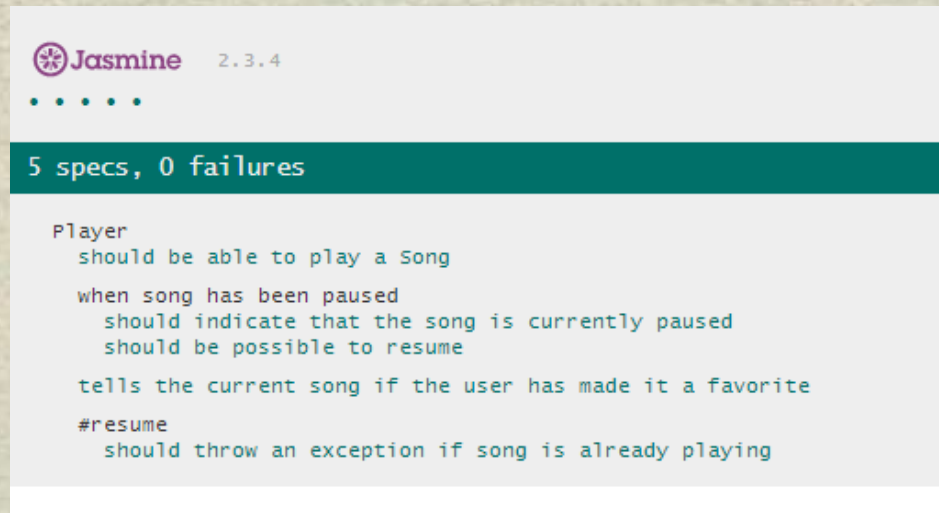
16. 코드 품질도구 jshint

- 설치 : `npm install --save-dev gulp-jasmine`
- 사이트 : <https://github.com/sindresorhus/gulp-jasmine>
- 용도 : 유닛 테스트

- 참고자료 : Outsider님의 자스민 자료([링크](#))를 참고하였습니다. Outsider님의 자료는 또 Evan Hahn 이라는 분의 튜토리얼([링크](#))을 참고하셨다고 합니다. CCL에 의한 표시함.
- 아...이왕할거-_-잠시 jasmine을 알아보도록 하겠습니다. 이곳에서 재스민을 다운받아쥬보겠습니다.
<https://github.com/jasmine/jasmine/releases>
- 압축을 풀어주시구요 돌려보시면 다음과 같은 화면이 나옵니다.

17. 테스트 도구 **jasmine**

- SpecRunner를 실행시켜서 다음과 같이 Javascript 테스트 결과부터를 보시다



src/Players.js

```

1 function Player() {
2 }
3 Player.prototype.play = function(song) {
4   this.currentlyPlayingSong = song;
5   this.isPlaying = true;
6 };
7
8 Player.prototype.pause = function() {
9   this.isPlaying = false;
10 };
11

```

검증

spec/PlayerSpec.js

```

describe("Player", function() {
  var player;
  var song;

  beforeEach(function() {
    player = new Player();
    song = new Song();
  });

  it("should be able to play a Song", function() {
    player.play(song);
    expect(player.currentlyPlayingSong).toEqual(song);

    //demonstrates use of custom matcher
    expect(player).toBePlaying(song);
  });
});

```

describe에서 이번 테스트할 것 정의하고

beforeEach에서 미리 준비를 하고

it에서 테스트를 돌리는 구조인듯?

17. 테스트 도구 jasmine 이런 구조?

- 이 슬라이드는 Jasmine을 집중적으로 다루는 문서가 아니므로 이정도만 짚고 넘어가겠습니다. 자세한 문서는 아웃사이드더님 블로그 [\(링크\)](#) 와 <http://jasmine.github.io/> 을 참조해주세요
- 여기서 해볼 것은 jasmine에 나온 javascript와 spec을 gulp쪽에도 옮겨보고, 헬로월드 정도의 테스트를 추가해서 jasmine테스트를 해보겠습니다.
- spec/test.js에 기존의 jasmine에 있던 src와 spec폴더에 있던 모든 내용들을 test.js에 옮겨주시고(...) 요런 내용도 추가해보겠습니다.

가장 기본적인 테스트랄까요?^^

```
function helloWorld() {  
    return "Hello world!";  
}  
  
describe("Hello world", function() {  
    it("says hello", function() {  
        expect(helloWorld()).toEqual("Hello world!");  
    });  
});
```

17. 테스트 도구 **jasmine**

- 다음과 같이 테스트가 잘 통과합니다. 6개의 검사를 잘 통과한 것 같군요..

```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> gulp jasmine
[16:45:53] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[16:45:53] Starting 'jasmine'...
.....
6 specs, 0 failures
Finished in 0 seconds
[16:45:53] Finished 'jasmine' after 46 ms
```

- 우선 말씀드릴 것은 제가 현상태는 gulp-test에 능숙한 상태는 아니구요. 우선 궁금 한 것이 있습니다. jasmine을 다운받아서 했을 때는 html파일을 실행시키고 분리된 javascript를 잘 실행시켰었는데.. 지금은 우선 급하게 파일을 모두 모아서 테스트를 해줬습니다.
- 뭐..생각해보면, concat하거나 따로 인식하고 연결시키는 방법이 있을 것같긴하지만;; 가상의 브라우저에서도 테스트를 할 수 있다는 방법이 귀동냥으로 들은 것 같습니다.. **phantom.js**를 해볼 시간입니다(?)

17. 테스트도구 **jasmine**

- 설치 : `npm install --save-dev gulp-jasmine2-phantomjs`
- 사이트 : <https://github.com/sandermak/gulp-jasmine2-phantomjs>
- 설명 : 가상브라우저를 통한 테스트

- 다음과 같이 설정을 해주고 spec폴더에 jasmine에서 봤던 것같은 SpecRunner.html 같은 파일을 만들어주겠습니다.

```
// 19. jasmine-phantom.js
var jasminePhantomJs = require('gulp-jasmine2-phantomjs');
var specFiles = 'SpecRunner.html'; //이부분은 제가 수정^^;
gulp.task('jasminePhantom', function() {
  return gulp.src(specFiles).pipe(jasminePhantomJs());
});
```

- 여기서 잠깐 주의하실게 phantom.js를 쓰시면서 jasmine에서 있던 js파일들을 그대로 써주면 안 되는 경우가 있습니다. 다음장을 보실까요

18. 가상브라우저 **phantom.js**

- jasmine을 다운받으시면서 폴더구조를 그대로 가져오시되, jasmine2-junit 이란 부분에서 다음과 같은 js파일로 변경을 해주어야 하는 부분들이 있습니다. 이점 유의해주시고, 필요한 js파일들 모두 복붙해주시고~, gulp 설정해준 명령어를 해보면.. 잘 동작하시는 것을 보실 수가 있습니다.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Jasmine Spec Runner v2.3.4</title>

  <link rel="shortcut icon" type="image/png" href="lib/jasmine-2.3.4/jasmine.png">
  <link rel="stylesheet" href="lib/jasmine-2.3.4/jasmine.css">

  <script src="lib/jasmine-2.3.4/jasmine.js"></script>
  <script src="lib/jasmine-2.3.4/jasmine-html.js"></script>

  <script src="lib/jasmine2-junit/jasmine2-junit.js"></script>
  <!-- 이름은 같은데 수정된 boot.js -->
  <script src="lib/jasmine2-junit/boot.js"></script>

  <!-- include source files here... -->
  <script src="src/Player.js"></script>
  <script src="src/Song.js"></script>

  <!-- include spec files here... -->
  <script src="spec/SpecHelper.js"></script>
  <script src="spec/PlayerSpec.js"></script>
```

```
PS C:\Users\arahansa\workspaceHTML\learn_frontend\gulp> gulp jasminePhantom
[17:58:40] Using gulpfile ~\workspaceHTML\learn_frontend\gulp\gulpfile.js
[17:58:40] Starting 'jasminePhantom'...
[17:58:42] Start running spec file: C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\spec\SpecRunner.html
[17:58:42] PhantomJS path: C:\Users\arahansa\workspaceHTML\learn_frontend\gulp\node_modules\phantomjs\bin\phantomjs.exe
Loading /Users/arahansa/workspaceHTML/learn_frontend/gulp/SpecRunner.html
Jasmine ran in 0.005 seconds
5 specs, 0 failures

[17:58:42] Finished 'jasminePhantom' after 1.15 s
```

18. 가상브라우저 phantom.js



찾다보니 이것저것 재밌는 기능들이 많아서 더
찾아 적고 싶고, 좀 더 production에서
쓸만한 설정들을 찾아 적고 싶었지만, 지금은
gulp자체의 문서 범위를 벗어나기도 하거니와
부처님 오신 날에 공개해야 할 슬라이드인데,
자꾸 시간을 끌게 되다보니...

지금은 이정도로 줄이고 다음 문서를
기대해보겠습니다.

다음엔 뭘 들고 찾아올까요..
즐거운 고민입니다. 이만 슬슬 마칩니다.

더 적고 싶은 게 많지만..(ㅠ)

- **여기도 많이 참고했음. LevelupTuts 유튜브 강좌**
<https://www.youtube.com/watch?v=wNIEK8qrb0M>

Joel님 강좌

- https://www.youtube.com/playlist?list=PLv1YUP7gO_vIROuRcGsDCNM-FUVgMYb_G
- **Gulp 설치관련**
<https://www.npmjs.com/package/gulp-install>


gulp 관련 더 볼만한 곳, 참고한 곳

- 많은 도움과 영감(?)을 주신 분들 감사합니다. ☺


본 슬라이드는 <http://ilikekillnerds.com/2014/11/10-highly-useful-gulp-js-plugins-for-a-super-ninja-front-end-workflow/> 를 기반으로 여기 나온 플러그인 깃헙들을 직접 방문하고 구글링하고 직접 해보면서 만들어봤습니다.


- 네이버D2세미나에서 사은품- 컵을 주신..
javajigi님께도 감사의 말씀을...;ㅁ;



 nipol 11:52 AM
https://www.youtube.com/playlist?list=PLv1YUP7gO_vIROuRcGsDCNM-FUVgMYb_G
In this series we will look at how to create a Gulp front-end build system.

이 시리즈도 추천합니다.
따라해 보면 별 것 없지만.
11:52 ★ 기초라면 충분하죠

 changhwaoh 9:55 PM
파일을 동시에 여러개 보고 편집해야한다 라는 메리트가 필요해지실거예요
물론 인텔리의 북마크 기능은 아쉽습니다
여러 장단점이있는듯해용 ㅎㅎ
겉프는 우선 필요목적부터보시는게
빌드가 주목적이면 빌드관련해선 다되어야겠죠

 Joel Longie @JoelLongie · 21분
@arahansa as long as you put the photo credit I'm fine with that. Good luck with your slides.

자세히

 Joel Longie @JoelLongie · 28분
@arahansa thanks :)

자세히

Special Thanks

- gulp 깃헙페이지에서는 다음과같은 샘플과 추천 플러그인을 얘기하는데....

Sample gulpfile.js

This file is just a quick sample to give you a taste of what gulp

```
var gulp = require('gulp');
var coffee = require('gulp-coffee');
var concat = require('gulp-concat');
var uglify = require('gulp-uglify');
var imagemin = require('gulp-imagemin');
var sourcemaps = require('gulp-sourcemaps');
var del = require('del');

var paths = {
  scripts: ['client/js/**/*.coffee', '!client/external/'],
  images: 'client/img/**/*.png'
};

// Not all tasks need to use streams
// A gulpfile is just another node program and you can
gulp.task('clean', function(cb) {
  // You can use multiple globbing patterns as you would
  del(['build'], cb);
});

gulp.task('scripts', ['clean'], function() {
  // Minify and copy all JavaScript (except vendor scri
  // with sourcemaps all the way down
  return gulp.src(paths.scripts)
    .pipe(sourcemaps.init())
```

Incremental Builds

We recommend these plugins:

- [gulp-changed](#) - only pass through changed files
- [gulp-cached](#) - in-memory file cache, not for operation on sets of files
- [gulp-remember](#) - pairs nicely with gulp-cached
- [gulp-newer](#) - pass through newer source files only, supports many:1 source:dest

앗참, gulp 홈페이지에서 추천하는.. < 66 >



-_-예... gulp 도 참..많네요;; 우선은 여기까지;;
(짤방: 이말년 시리즈 : [https://mirror.enha.kr/wiki/고만해, 미친놈들아!](https://mirror.enha.kr/wiki/고만해,미친놈들아!))

THANK YOU !



arahansa

페북 : <https://fb.com/me.adunhansa>
트위터 : <https://twitter.com/arahansa>
블로그 : <http://adunhansa.tistory.com/>
사이트 : <http://arahansa.com>