

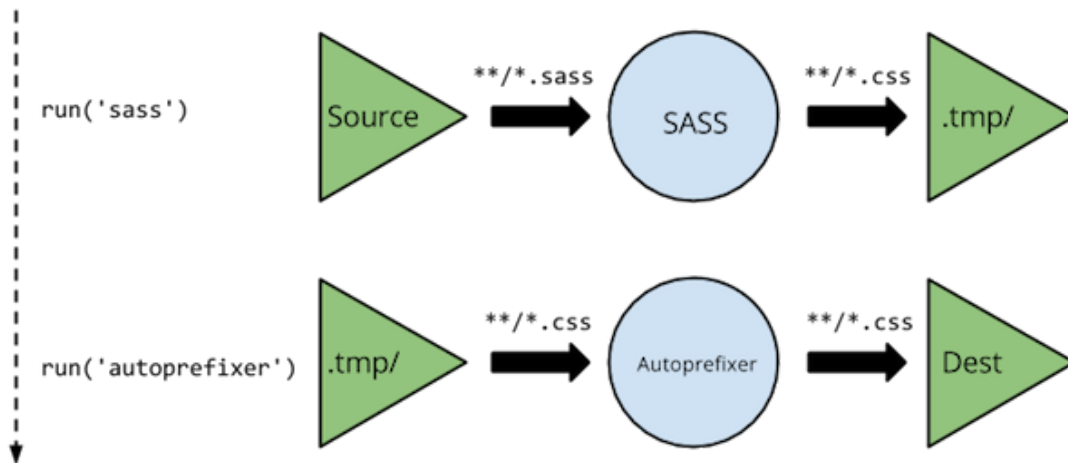
# Gulp

## Materials

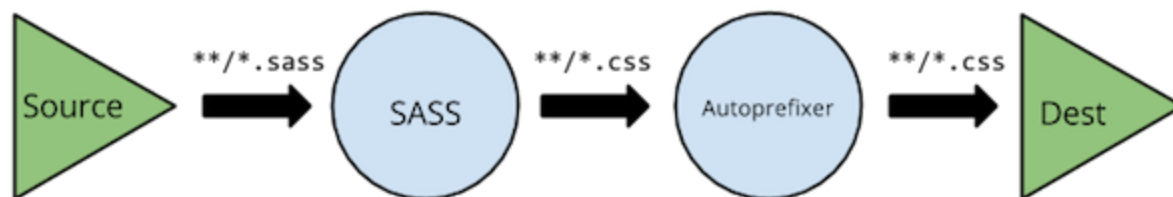
- Official Documentation
  - Official Website: <http://gulpjs.com/>
  - Github Repository: <https://github.com/gulpjs/gulp>
- Recipes
  - <https://github.com/gulpjs/gulp/tree/master/docs/recipes>
- Courses
  - Web Tooling & Automation - Gulp, Sass, and BabelJS, Oh My!
- Articles
  - Grunt vs Gulp - Beyond the Numbers
  - Gulp 4: The new task execution system - gulp.parallel and gulp.series
  - Why I Left Gulp and Grunt for npm Scripts
- Videos
  - Totally Tooling Tips: Intro to Gulp 4 (S3 Mini Tip #1)

## Grunt vs Gulp

*In Grunt, we must write intermediary files to disk*



*In Gulp, we pipe the intermediary files in-memory to other streams*



Article: [Grunt vs Gulp - Beyond the Numbers](#)

## Node.js and Javascript

Node.js version is 4.2.2

For more information, please visit: [Node.js Long-term Support Working Group](#)

We will use EcmaScript 6 with <https://babeljs.io/>

## Gulp

### Gulp Documentation

Gulp is a toolkit that will help you automate painful or time-consuming tasks in your development workflow. For web development (if that's your thing) it can help you by doing CSS preprocessing, JS transpiling, minification, live reloading, and much more.

Integrations are built into all major IDEs and people are loving gulp across PHP, .NET, Node.js, Java, and more. With over 1700 plugins (and plenty you can do without plugins), gulp lets you quit messing with build systems and get back to work.

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

### Install gulp globally

```
npm install --global gulp
```

### Install gulp in your project devDependencies

```
npm install --save-dev gulp
```

### Usage

```
import gulp from 'gulp';

gulp.task('default', function() {
  // Place code for you default task here.
});
```

## Gulp 4.0

### Usage

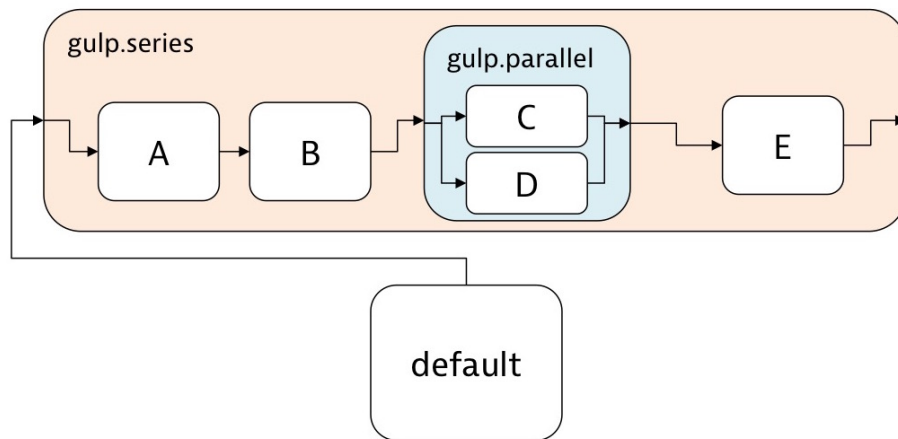
```
# uninstal gulp 3 cli & local gulp
npm uninstall gulp-cli -g
cd {project-folder}
npm install uninstall gulp

# Install gulp 4 CLI
npm install gulpjs/gulp-cli#4.0 -g

# install gulp 4 locally in your project
npm install gulpjs/gulp.git#4.0 --save-dev
```

Gulp.series and Gulp.parallel

- `gulp.series` for sequential execution
- `gulp.parallel` for parallel execution.



Article: [Gulp 4: The new task execution system - gulp.parallel and gulp.series](#)

## Gulp Babel

### Gulp Babel Documentation

Use next generation JavaScript, today, with Babel

Repository: <https://github.com/babel/gulp-babel>

### Installation

```
npm install babel-core gulp-babel babel-preset-es2015  
babel-preset-es2016 --save-dev
```

After please read follow article: [Using ES6 with gulp.](#)

We need to create a new file:

### Create a new file - babelrc

```
touch .babelrc
```

And add the following to the file:

### Babelrc Content

```
{  
  "presets": [ "es2016", "es2015" ]  
}
```

## Gulp Util

### Gulp Util Documentation

Utility functions for gulp plugins

Repository: <https://github.com/gulpjs/gulp-util>

### Installation

```
npm install --save-dev gulp-util
```

### Usage

```
import gulp from 'gulp';
import gutil from 'gulp-util';

gutil.log('stuff happened', 'Really it did',
gutil.colors.magenta('123'));
gutil.beep();

gutil.replaceExtension('file.coffee', '.js'); // file.js

var opt = {
  name: 'todd',
  file: someGulpFile
};
gutil.template('test <%= name %> <%= file.path %>', opt) // test todd
/js/hi.js
```

## Gulp Imagemin

### Gulp Imagemin Documentation

Minify PNG, JPEG, GIF and SVG images with [imagemin](#)

Repository: <https://github.com/sindresorhus/gulp-imagemin>

### Installation

```
npm install --save-dev imagemin-pngquant # imagemin pngquant
npm install --save-dev gulp-imagemin # gulp imagemin
```

### Usage

```
import gulp from 'gulp';
import gutil from 'gulp-imagemin';
import pngquant from 'imagemin-pngquant';

gulp.task('default', function () {
  return gulp.src('src/images/*')
    .pipe(imagemin({
      progressive: true,
      svgoPlugins: [{removeViewBox: false}],
      use: [pngquant()]
    }))
    .pipe(gulp.dest('dist/images'));
});
```

## Gulp Concat

### Gulp Concat Documentation

Concatenates files

Repository: <https://github.com/contra/gulp-concat>

### Installation

```
npm install --save-dev gulp-concat
```

### Usage

```
import gulp from 'gulp';
import concat from 'gulp-concat';

gulp.task('scripts', function () {
  return gulp.src('./lib/*.js')
    .pipe(concat('all.js'))
    .pipe(gulp.dest('./dist/'));
});
```

## Gulp Order

### Gulp Order Documentation

The gulp plugin `gulp-order` allows you to reorder a stream of files using the same syntax as of `gulp.src`.

Repository: <https://github.com/sirlantis/gulp-order>

### Installation

```
npm install --save-dev gulp-order
```

### Usage

```
import gulp from 'gulp';
import order from 'gulp-order';
import coffee from 'gulp-coffee';
import concat from 'gulp-concat';

gulp.task('scripts', function () {
  return gulp.src("**/*.coffee")
    .pipe(coffee())
    .pipe(gulp.src("**/*.js")) // gulp.src passes through input
    .pipe(order([
      "vendor/js1.js",
      "vendor/**/*.js",
      "app/coffee1.js",
      "app/**/*.js"
    ]))
    .pipe(concat("all.js"))
    .pipe(gulp.dest("dist"));
});
```

## Gulp Wrap

### Gulp Wrap Documentation

A [gulp](#) plugin to wrap the stream contents with a [lodash](#) template.

Repository: <https://github.com/adamayres/gulp-wrap>

### Installation

```
npm install --save-dev gulp-wrap
```

### Usage

```
import gulp from 'gulp';
import wrap from 'gulp-wrap';

gulp.task('scripts', function () {
  return gulp.src("./src/*.json")
    .pipe(wrap('Hello, <%= contents.title %>, have a <%=
contents.adjective %> day.'))
    .pipe(gulp.dest("./dist"));
});
```

## StreamQueue

### StreamQueue Documentation

StreamQueue pipe the queued streams one by one in order to preserve their content order.

Repository: <https://github.com/nfroidure/StreamQueue>

### Installation

```
npm install --save-dev streamqueue
```

### Usage

```
import streamqueue from 'streamqueue';

let queue = new streamqueue();
queue.queue(
  Fs.createReadStream('input.txt'),
  Fs.createReadStream('input2.txt'),
  Fs.createReadStream('input3.txt')
);
queue.done();

queue.pipe(process.stdout);
```

## Run-Sequence

### Run-Sequence Documentation

Runs a sequence of gulp tasks in the specified order. This function is designed to solve the situation where you have defined run-order, but choose not to or cannot use dependencies.

Repository: <https://github.com/OverZealous/run-sequence>

### Installation

```
npm install --save-dev run-sequence
```

## Usage

```
import gulp from 'gulp';
import runSequence from 'run-sequence';
import del from require('del');

// This will run in this order:
// * build-clean
// * build-scripts and build-styles in parallel
// * build-html
// * Finally call the callback function
gulp.task('build', function(callback) {
  runSequence('build-clean',
    ['build-scripts', 'build-styles'],
    'build-html',
    callback);
});

// configure build-clean, build-scripts, build-styles, build-html as you
// wish, but make sure they either return a stream or handle the
// callback
// Example:
gulp.task('build-clean', function(callback) {
  del([BUILD_DIRECTORY], callback);
  //          ^^^^^^^
  //   This is the key here, to make sure tasks run asynchronously!
});

gulp.task('build-scripts', function() {
  return gulp.src(Scripts_SRC).pipe(...);
  //   ^^^^^
  //   This is the key here, to make sure tasks run asynchronously!
});
```

## Del

### Del Documentation

Delete files and folders using [globs](#)

Repository: <https://github.com/sindresorhus/del>

## Installation

```
npm install --save-dev del
```



### Usage

```
import gulp from 'gulp';
import concat from 'gulp-concat';
import del from 'del';

/**
 * Task will delete concatenation temp files.
 */
gulp.task('concat:clean', function() {
  var javascript_files = [
    './assets/dist/javascript/libraries.min.js',
    './assets/dist/javascript/project.min.js'
  ];

  var css_files = [
    './assets/dist/styles/libraries.min.css',
    './assets/dist/styles/project.min.css'
  ];

  var garbage_files = javascript_files.concat(css_files);

  del(garbage_files, function (error, paths) {
  });
});
```

## Gulp uglify

### Gulp uglify Documentation

Minify JavaScript with UglifyJS2.

Repository: <https://github.com/terinjokes/gulp-uglify>

### Installation

```
npm install --save-dev gulp-uglify
```

### Usage

```
import gulp from 'gulp';
import uglify from 'gulp-uglify';

gulp.task('compress', function() {
  return gulp.src('lib/*.js')
    .pipe(uglify())
    .pipe(gulp.dest('dist'));
});
```

## Gulp minify css

### Gulp minify css Documentation

gulp plugin to minify CSS, using [clean-css](#)

Repository: <https://github.com/terinjokes/gulp-uglify>

#### Installation

```
npm install --save-dev gulp-minify-css
```

#### Usage

```
import gulp from 'gulp';
import minifyCss from 'gulp-minify-css';

gulp.task('minify-css', function() {
  return gulp.src('styles/*.css')
    .pipe(minifyCss({compatibility: 'ie8'}))
    .pipe(gulp.dest('dist'));
});
```

## Gulp notify

### Gulp notify Documentation

notification plugin for [gulp](#)

Repository: <https://github.com/mikaelbr/gulp-notify>

#### Installation

```
npm install --save-dev gulp-notify
```

## Yargs

### Yargs Documentation

Yargs be a node.js library fer hearties tryin' ter parse optstrings. With yargs, ye be havin' a map that leads straight to yer treasure! Treasure of course, being a simple option hash.

Repository: <https://github.com/bcoe/yargs>

#### Installation

```
npm install yargs
```

### Usage

```
# plunder.js:

import yargs from 'yargs';

#!/usr/bin/env node
const argv = yargs.argv;

if (argv.ships > 3 && argv.distance < 53.5) {
  console.log('Plunder more riffiwobbles!');
}
else {
  console.log('Retreat from the xuptumblers!');
}
```

### Execution

```
$ ./plunder.js --ships=4 --distance=22
Plunder more riffiwobbles!

$ ./plunder.js --ships 12 --distance 98.7
Retreat from the xuptumblers!
```

## Gulp watch

### Gulp watch Documentation

File watcher that uses super-fast [chokidar](#) and emits vinyl objects.

Repository: <https://github.com/floatdrop/gulp-watch>

### Installation

```
npm install --save-dev gulp-watch
```

### Usage

```
import gulp from 'gulp';
import watch from 'gulp-watch';

gulp.task('stream', function () {
  return gulp.src('css/**/*.css')
    .pipe(watch('css/**/*.css'))
    .pipe(gulp.dest('build'));
});

gulp.task('callback', function (cb) {
  watch('css/**/*.css', function () {
    gulp.src('css/**/*.css')
      .pipe(watch('css/**/*.css'))
      .on('end', cb);
  });
});
```

## Gulp sourcemaps

### Gulp sourcemaps Documentation

Repository: <https://github.com/floridoo/gulp-sourcemaps>

### Installation

```
npm install --save-dev gulp-sourcemaps
```

### Usage

```
import gulp from 'gulp';
import uglify from 'gulp-uglify';
import sourcemaps from 'gulp-sourcemaps';

gulp.task('javascript', function() {
  gulp.src('src/**/*.js')
    .pipe(sourcemaps.init())
    .pipe(uglify())
    .pipe(sourcemaps.write('/', { addComment: false }))
    .pipe(gulp.dest('dist'));
});
```