

## ファイル監視



```
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']);
});
```

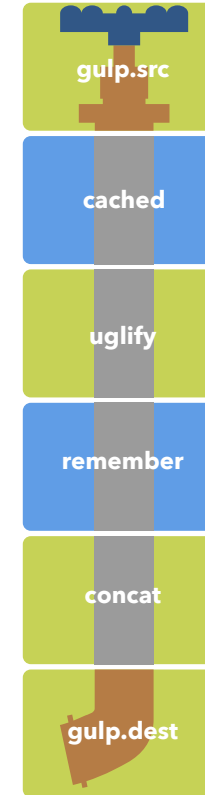
## 複数の保存先

```
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/'));
});
```



## 増分リビルド



```
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/'));
});
```

## タスクの依存関係

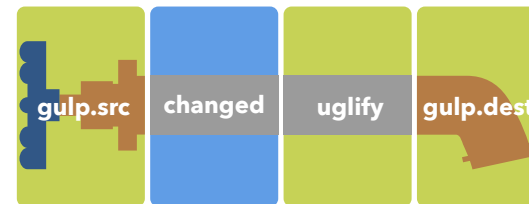
```
gulp.task('css', ['font', 'less'], function(){
  // do something after font and less
});
```

```
$ 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
```

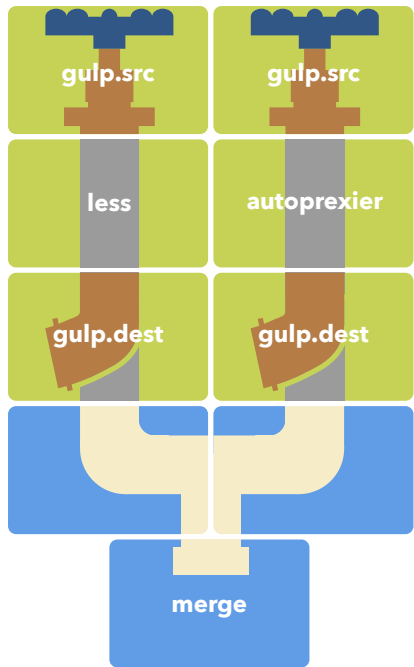


## 更新ファイルだけ処理

```
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/'));
});
```

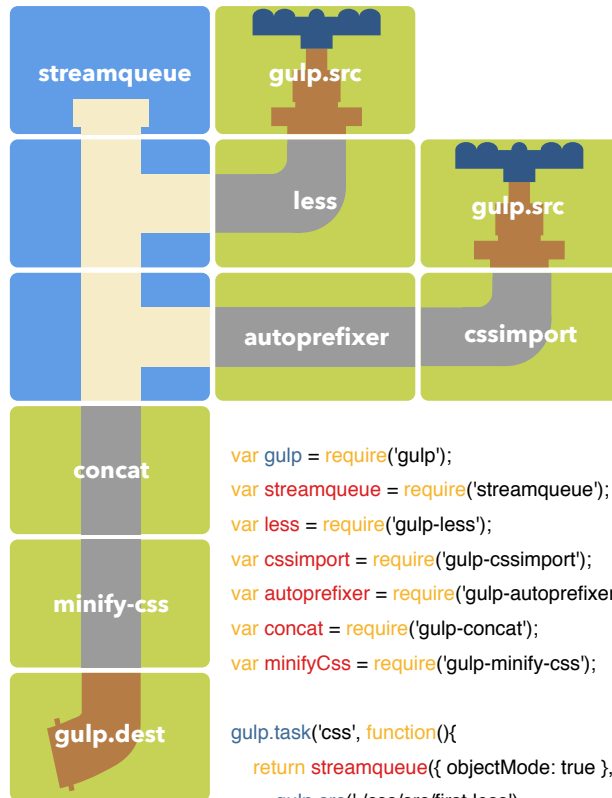
## 非同期タスク



```
var gulp = require('gulp');
var merge = require('merge-stream');
var less = require('gulp-less');
var autoprefixer = require('gulp-autoprefixer');
```

```
gulp.task('css', function(){
  return merge(
    gulp.src('./css/src/first.less')
      .pipe(less())
      .pipe(gulp.dest('./css/')),
    gulp.src('./css/src/second.css')
      .pipe(autoprefixer())
      .pipe(gulp.dest('./css/'))
  );
});
```

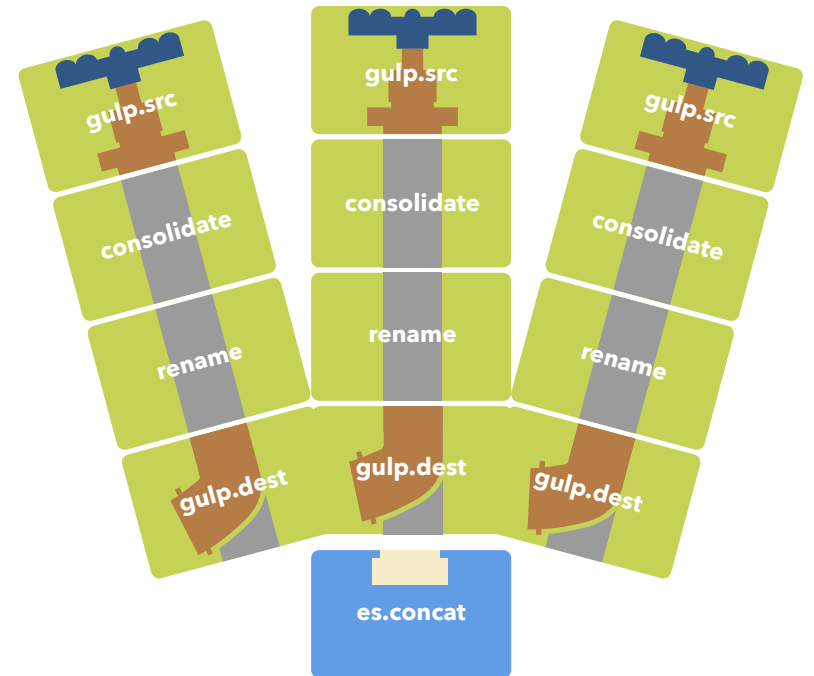
## シリアルタスク



```
var gulp = require('gulp');
var streamqueue = require('streamqueue');
var less = require('gulp-less');
var cssimport = require('gulp-cssimport');
var autoprefixer = require('gulp-autoprefixer');
var concat = require('gulp-concat');
var minifyCss = require('gulp-minify-css');
```

```
gulp.task('css', function(){
  return streamqueue({ objectMode: true },
    gulp.src('./css/src/first.less')
      .pipe(less()),
    gulp.src('./css/src/second.css')
      .pipe(cssimport())
      .pipe(autoprefixer('last 2 versions'))
      .pipe(concat('app.css'))
      .pipe(minifyCss())
      .pipe(gulp.dest('./css/'))
  );
});
```

## タスクの配列



```
var gulp = require('gulp');
var es = require('event-stream');
var consolidate = require('gulp-consolidate');
var rename = require('gulp-rename');
```

```
data = [
  {
    name: 'apple',
    title: 'Apple Cake',
    attrs: some_data
  },
  {
    name: 'orange',
    title: 'Orange Cookie',
    attrs: some_data
  }
];
```

```
gulp.task('page', function(){
  var tasks = data.map(function(entry){
    return gulp.src('./templates/a.js')
      .pipe(consolidate({
        title: entry.title,
        attrs: entry.attrs
      }))
      .pipe(rename({
        basename: entry.name,
        extname: '.html'
      }));
  });
  return es.concat.apply(null, tasks);
});
```