

Accelerated Front-end Development with

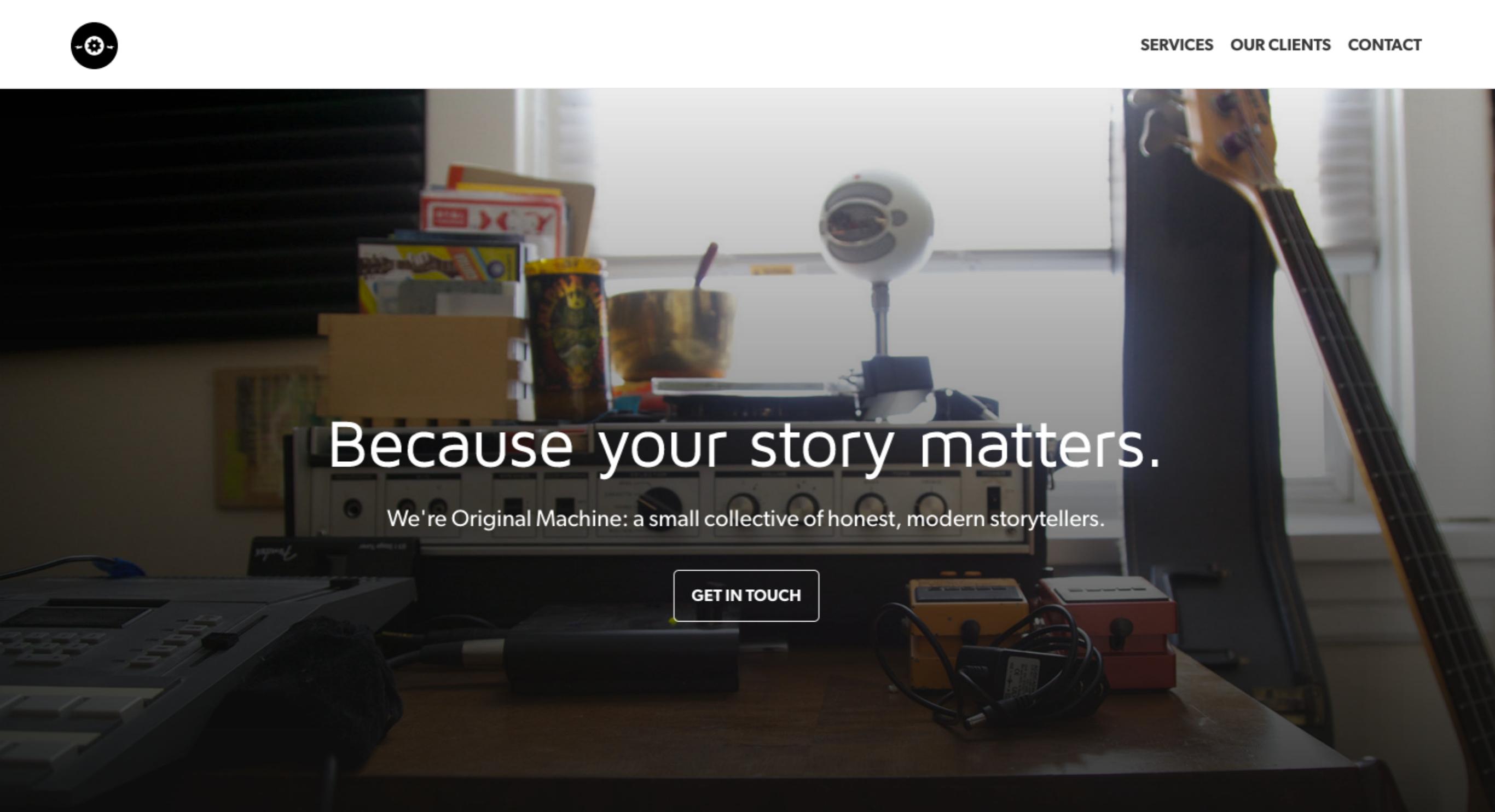


Who is this guy?



@nicholaswyoung

radio host
designer
photographer
extraordinarily busy person

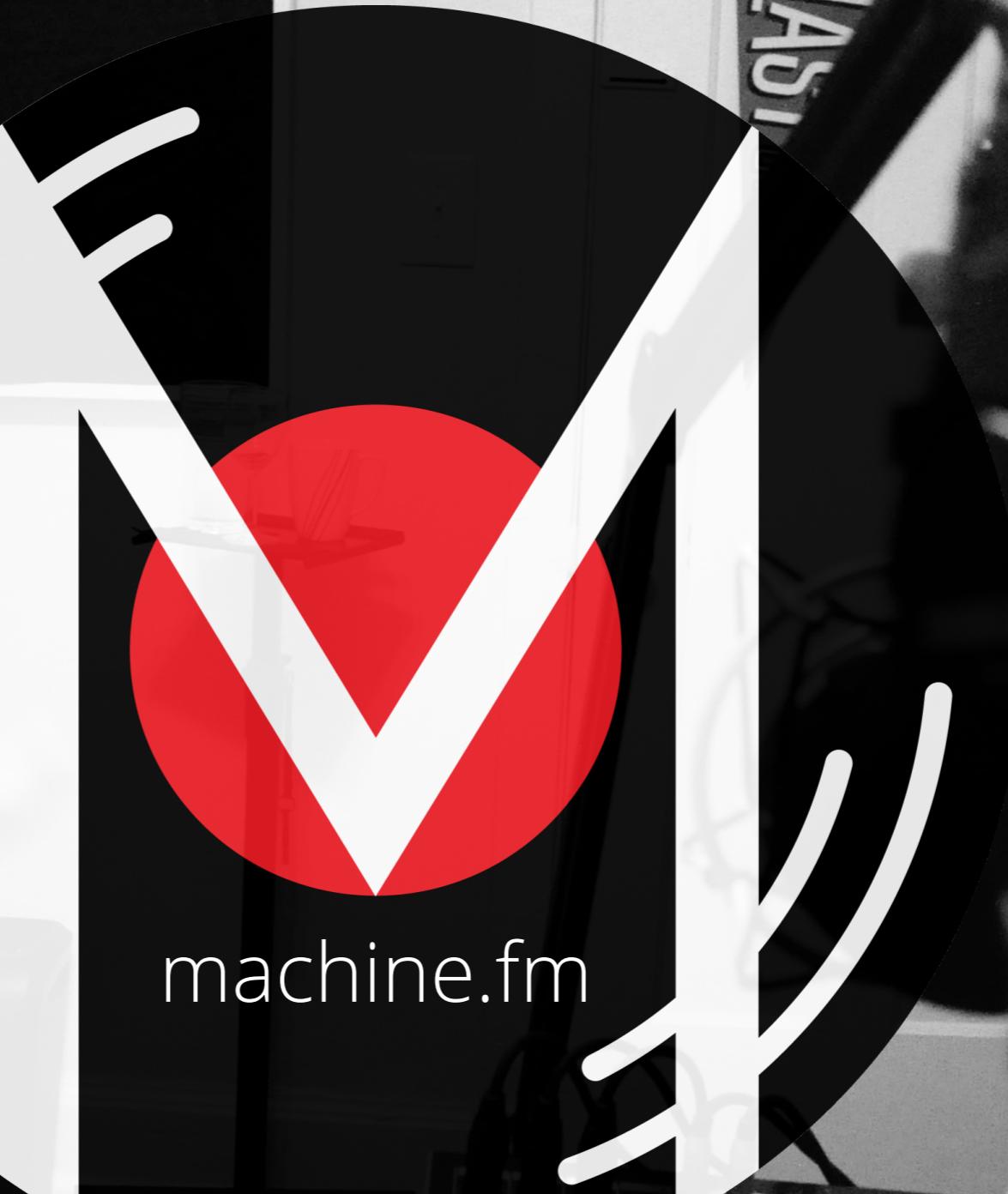


Because your story matters.

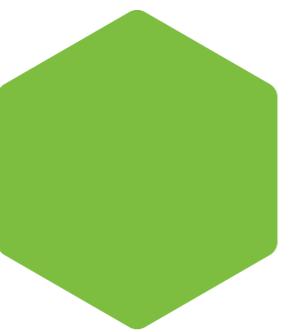
We're Original Machine: a small collective of honest, modern storytellers.

[GET IN TOUCH](#)

originalmachine.com



machine.fm







<http://nodejs.org/download>

brew install node

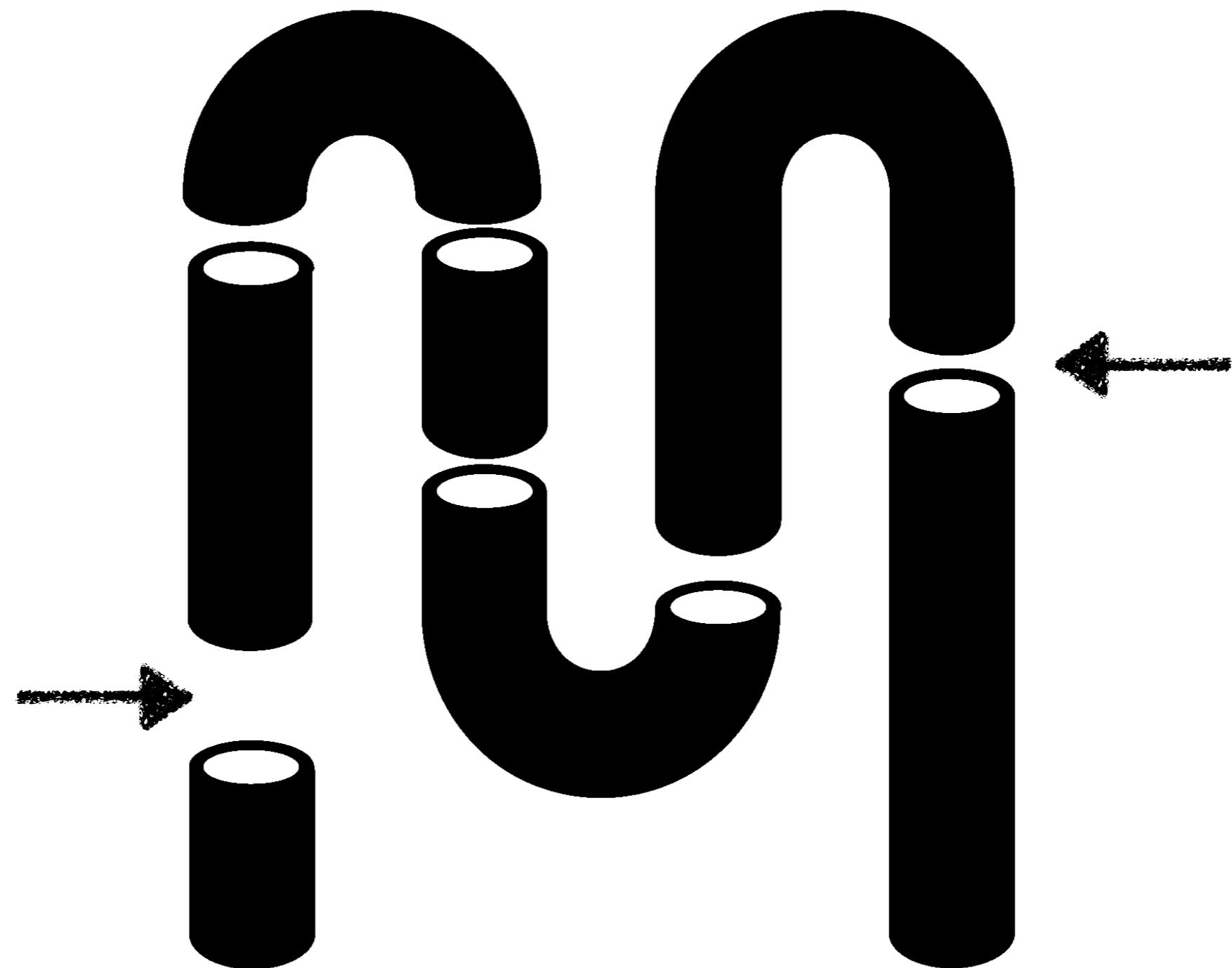
scoop install nodejs

```
git clone git://github.com/nicholaswyoung/cwc-2015.git
```

```
<div id='wrap' style='height: 100%; width: 100%;'></div>
```

Building a modern front-end
application can be difficult.

asset versioning
testing CSS pre-processing
concatenation / linting
templating
auto-reloading





```
module.exports = function (grunt) {
  grunt.initConfig({
    jshint: {
      files: [
        'Gruntfile.js',
        'src/**/*.js',
        'test/**/*.js'
      ],
      watch: {
        files: ['<%= jshint.files %>'],
        tasks: ['jshint']
      }
    });
  });

  grunt.loadNpmTasks('grunt-contrib-jshint');
  grunt.loadNpmTasks('grunt-contrib-watch');

  grunt.registerTask('default', ['jshint']);
};
```



- Config isn't the clearest.
- Doing anything of consequence requires a plugin.
- Every bundle of files must be written to a tempfile before the final output.

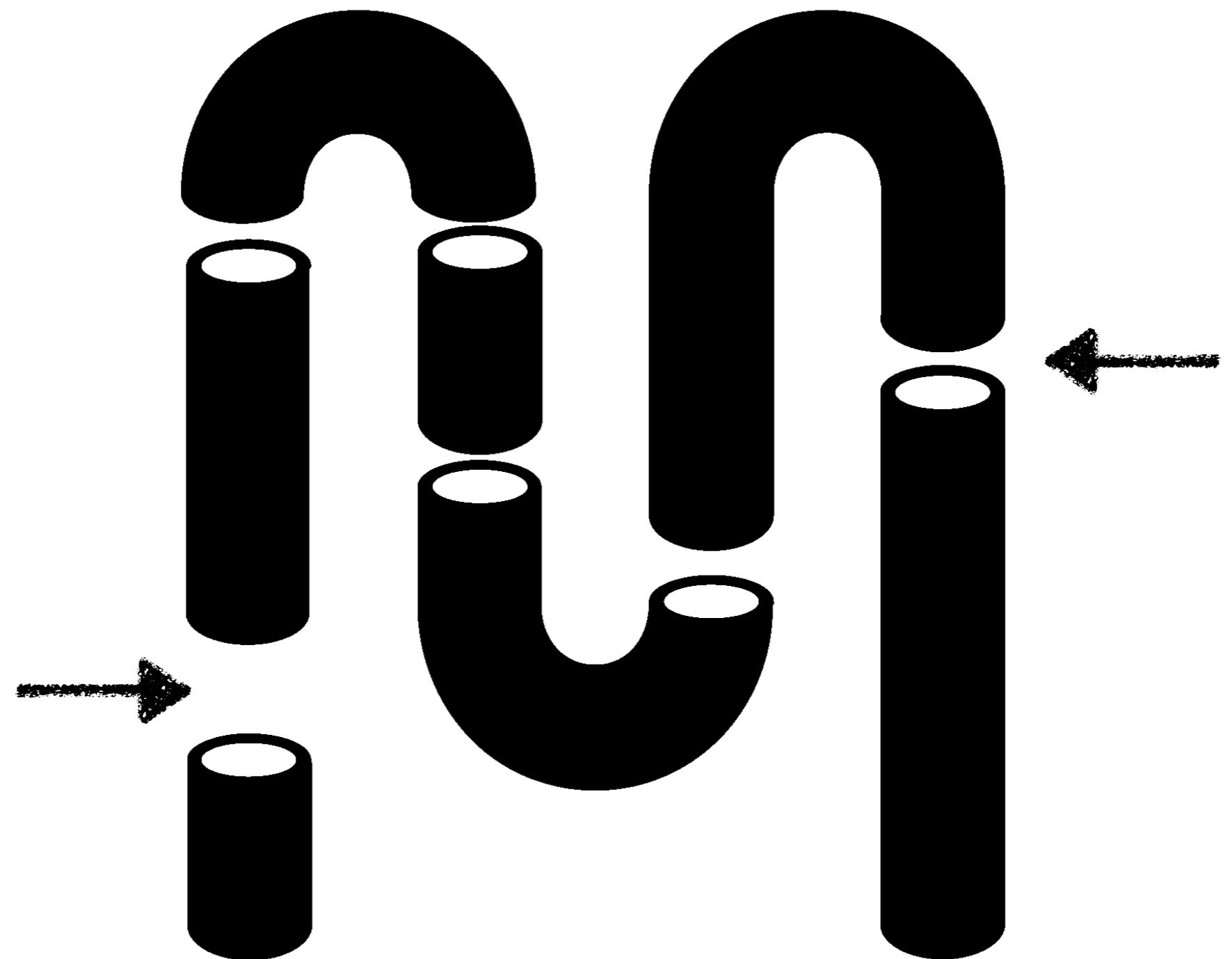


```
module.exports = function (grunt) {
  grunt.initConfig({
    jshint: {
      files: [
        'Gruntfile.js',
        'src/**/*.js',
        'test/**/*.js'
      ],
      watch: {
        files: ['<%= jshint.files %>'],
        tasks: ['jshint']
      }
    });
  });

  grunt.loadNpmTasks('grunt-contrib-jshint');
  grunt.loadNpmTasks('grunt-contrib-watch');

  grunt.registerTask('default', ['jshint']);
};
```

- Processor Cache
- RAM
- Disk I/O
- Network I/O



```
var Responder = require('./')
, response = new Responder();

response.on('data', function (data) {
  console.log(data.toString());
});

response.write('Hey CWC JavaScript Geeks!');
```

```
var fs      = require('fs')
,   app    = require('express')();

app.get('/', function (req, res) {
  res.type('html');
  fs.createReadStream('./index.html').pipe(res);
});

app.listen(9292);
```

```
var fs          = require('fs')
, oppressor = require('oppressor')
, app      = require('express')();

app.get('/', function (req, res) {
  res.type('html');
  fs.createReadStream('./index.html')
    .pipe(oppressor(req)).pipe(res);
}) ;

app.listen(9292);
```

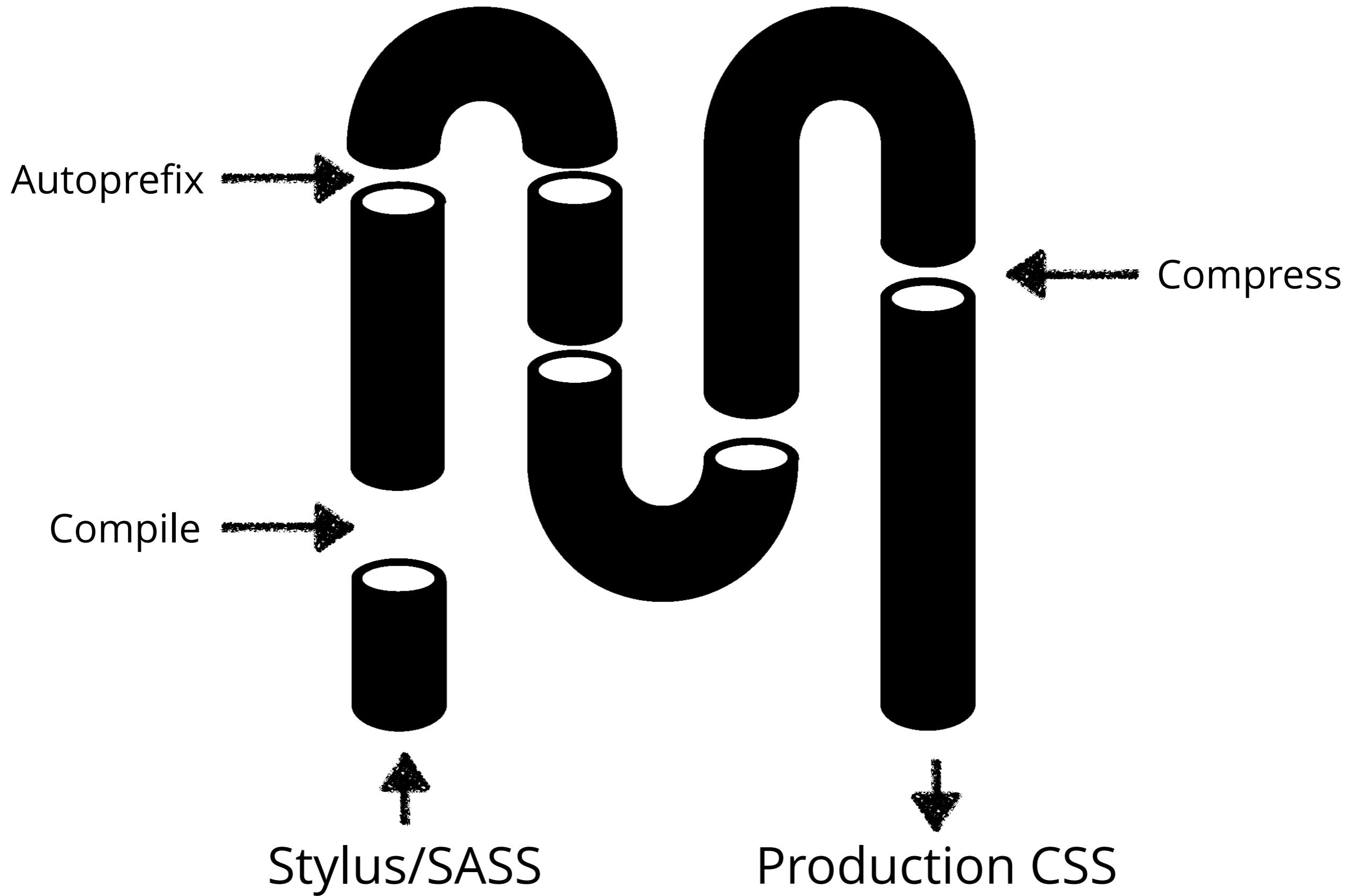
“Streams come to us from the earliest days of unix and have proven themselves over the decades as a dependable way to compose large systems out of small components that do one thing well.

You can then plug the output of one stream to the input of another and use libraries that operate abstractly on streams to institute higher-level flow control.”

- @substack

“We should have some ways of connecting programs like garden hose — screw in another segment when it becomes necessary to massage data in another way. This is the way of IO also.”

- DOUGLAS MCILROY



```
var gulp = require('gulp')

gulp.task('css', function () {
  return gulp.src('src/css/app.styl')
    .pipe($.stylus())
    .pipe(gulp.dest('dist')));
}) ;
```

```
var gulp = require('gulp')

gulp.task('css', function () {
  return gulp.src('src/css/app.styl') ← Input
    .pipe($.stylus())
    .pipe(gulp.dest('dist')); ← Output
});
```



```
module.exports = function (grunt) {
  grunt.initConfig({
    jshint: {
      files: [
        'Gruntfile.js',
        'src/**/*.js',
        'test/**/*.js'
      ],
      },
      watch: {
        files: ['<%= jshint.files %>'],
        tasks: ['jshint']
      }
    });
  grunt.loadNpmTasks('grunt-contrib-jshint');
  grunt.loadNpmTasks('grunt-contrib-watch');

  grunt.registerTask('default', ['jshint']);
};
```



```
var gulp  = require('gulp')
,  jshint = require('gulp-jshint');

gulp.task('jshint', function () {
  return gulp.src([
    'Gruntfile.js',
    'src/**/*.js',
    'test/**/*.js'
  ]).pipe(jshint())
});

gulp.task('watch', function () {
  gulp.watch([
    'Gruntfile.js',
    'src/**/*.js',
    'test/**/*.js'
  ], ['jshint']);
})

gulp.task('default', ['jshint', 'watch']);
```

```
var gulp    = require('gulp')
, jshint = require('gulp-jshint');

var inputs = ['Gruntfile.js', 'src/**/*.js', 'test/**/*.js'];

gulp.task('jshint', function () {
  return gulp.src(inputs).pipe(jshint())
});

gulp.task('watch', function () {
  gulp.watch(inputs, ['jshint']);
})

gulp.task('default', ['jshint', 'watch']);
```



[gulpjs.com/plugins](#)

gulp.js

Search 1311 plugins...

add-stream

Append the contents of one stream onto another.

gulpfriendly stream append add concat

amd-optimize

An AMD (i.e. RequireJS) optimizer that's stream-friendly. Made for gulp. (WIP)

gulpplugin gulpfriendly

auto-plug

Auto-require plugin packages by prefix. (for i.e. Gulp, Grunt or other heavy plugin-dependant packages)

```
var gulp = require('gulp');

gulp.task('default', function (done) {
  //...
});
```



LIVE CODING

- Don't touch the disk unless you want to.
- There's only one spec to worry about: through streams.
- It's fast.



Q & A

ask me anything



@nicholaswyoung

nicholaswyoung.com

OriginalMachine.com

Mixdown.co

Machine.FM