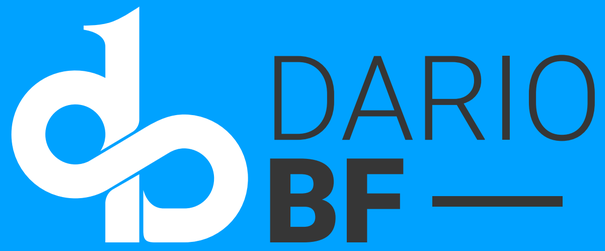
PostCSS

Herramientas modernas para el desarrollo de temas de WordPress

WordPress Las Palmas de Gran Canaria



Diseñador UI/UX
Formador
Dev especializado en WordPress
ActualidadBlog

@DarioBF

dariobf.com/podcast

Lo que NO es PostCSS

Un pre-processor
Un post-procesador
La sintaxis del futuro
Una herramienta de limpieza u optimización

Lo que SI es PostCSS

Una herramienta que nos permite transformar
CSS a través de JavaScript

Gulp vs Grunt vs Webpack

Vs Broccoli vs Brunch vs ENB vs Fly vs Stylus vs Meteor vs Duo vs Connect/Express

	Browserify	Grunt	Gulp	Webpack
Description	Browser-side require() the node way	The Java Script Task Runner	The streaming build system	Module bundler
Created	Feb, 2011	Jan. 2012	Jul, 2013	Mar, 2012
Version Average	every 6 days	every month	every month	every 4 days
Maintainers	40	4	2	3
Dependencies	48	17	13	25
Monthly downloads	2,261,042	2,038,947	3,387,902	14,960,064
Open issues	322	135	20	491
Open pull requests	30	20	0	30
GitHub stars	12,115	11,837	29,982	42,790
Subscribers	318	570	1,239	1,542
Forks	1,059	1,544	4,353	5,407

Crédito


CSS IS
AWESOME

Ventajas

CSS Simple
Versátil
Modularización (lo mejor y lo peor de PostCSS)
Comunidad
Velocidad

Tiempos de ejecución

PostCSS:	35 ms	
Rework:	38 ms	(1.1 times slower)
LibSass sync:	82 ms	(2.3 times slower)
Stylus:	87 ms	(2.5 times slower)
LibSass:	90 ms	(2.5 times slower)
Less:	91 ms	(2.6 times slower)
Dart Sass sync:	103 ms	(2.9 times slower)
Dart Sass:	169 ms	(4.8 times slower)
Stylecow:	199 ms	(5.6 times slower)

Instalación con Gulp

```
$ npm install --save-dev gulp-postcss
```

gulpfile.js

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


gulp.task('magiapotagia', function () {
    var processors = [
        //Aquí irán los diferentes plugins que vamos añadiendo
    ];
    return gulp.src('./src/*.css')
        .pipe(postcss(processors))
        .pipe(gulp.dest('./dist'));
});
```

Activamos sourcemaps

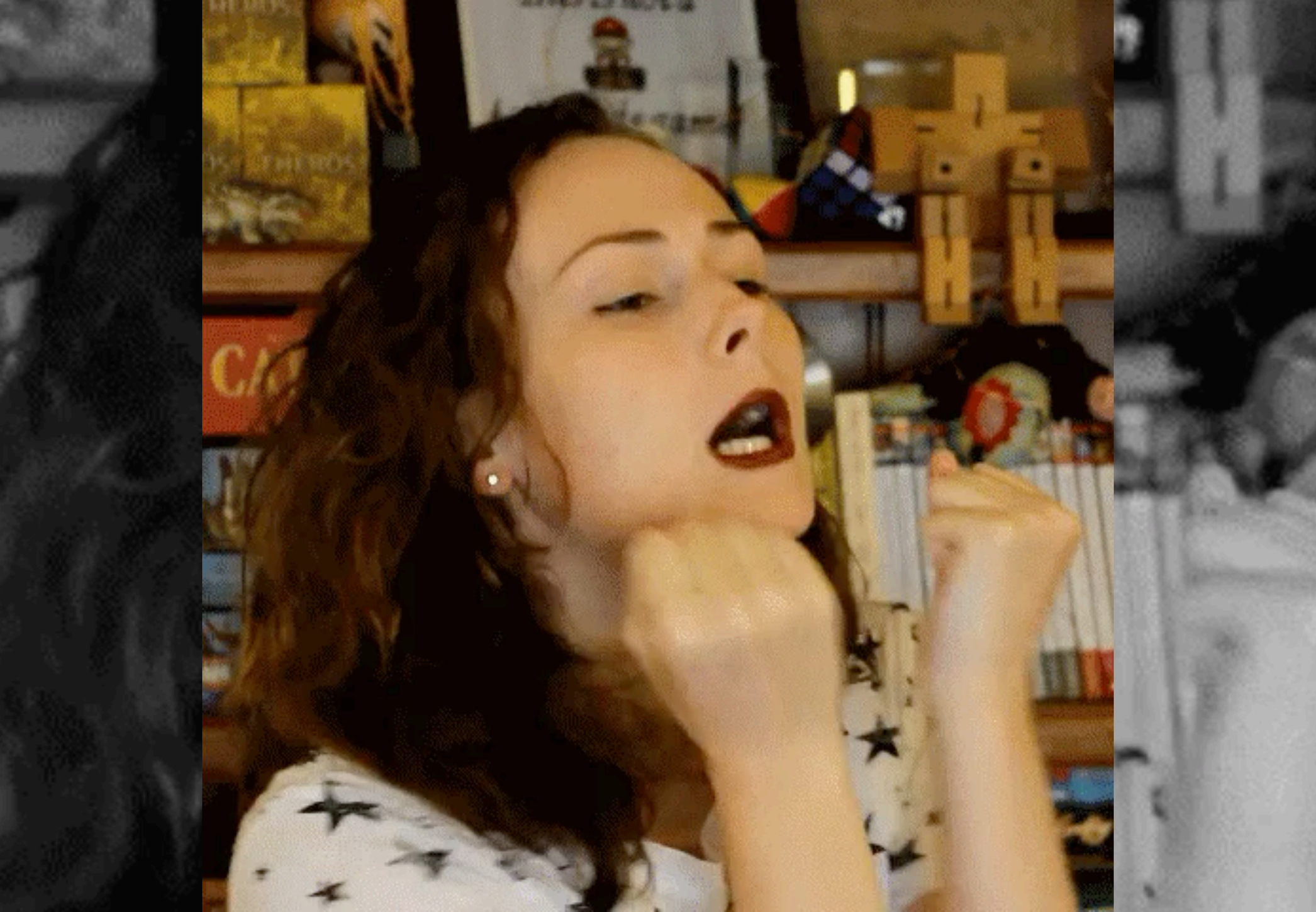
```
var postcss = require('gulp-postcss');

var gulp = require('gulp');

gulp.task('magiapotagia', function () {
    var processors = [
        //Aquí irán los diferentes plugins que vamos añadiendo
    ];
    return gulp.src('./src/*.css')
        .pipe(sourcemaps.init())
        .pipe(postcss(processors))
        .pipe(sourcemaps.write('.'))
        .pipe(gulp.dest('./dist'));
});
```

Plugins

postcss.parts



atImport

Utilizar @import tanto con archivos localhost
como node o web modules

Using PostCSS for Minification and Optimization
https://webdesign.tutsplus.com/tutorials/using-postcss-for-minification-and-optimization--cms-24568

PostCSS-import Repository
https://github.com/postcss/postcss-import

atImport

`css/input.css` :

```
/* can consume `node_modules`, `web_modules` or local modules */
@import "cssrecipes-defaults"; /* == @import "../node_modules/cssrecipes-defaults/index.css"; */
@import "normalize.css"; /* == @import "../node_modules/normalize.css/normalize.css"; */

@import "foo.css"; /* relative to css/ according to `from` option above */

@import "bar.css" (min-width: 25em);

body {
  background: black;
}
```

will give you:

```
/* ... content of ../node_modules/cssrecipes-defaults/index.css */
/* ... content of ../node_modules/normalize.css/normalize.css */

/* ... content of css/foo.css */

@media (min-width: 25em) {
/* ... content of css/bar.css */
}

body {
  background: black;
}
```

preCSS

Sintaxis al estilo SASS

https://github.com/jonathantneal/precss

PreCSS

```
$blue: #056ef0;
$column: 200px;

.menu {
  width: calc(4 * $column);
}

.menu_link {
  background: $blue;
  width: $column;
}

/* becomes */

.menu {
  width: calc(4 * 200px);
}

.menu_link {
  background: #056ef0;
  width: 200px;
}
```

Nested & Nesting

Anidamiento de SASS

https://github.com/postcss/postcss-nested

Nesting al estilo W3C:
http://tabatkins.github.io/specs/css-nesting/

autoprefixer

Prefijos para navegadores

https://github.com/postcss/autoprefixer

Algunos más

postcss-mixins
pxtorem
stylelint + reporter
cssnano
cssnext

@apply

```
/* input */
:root {
  --toolbar-theme: {
        background-color: rebeccapurple;
        color: white;
        border: 1px solid green;
  };
  --toolbar-title-theme: {
        color: green;
  };
}

.toolbar {
        @apply --toolbar-theme;
}

.toolbar-title {
        @apply --toolbar-title-theme;
}
```

```
/* output */
.toolbar {
        background-color: rebeccapurple;
        color: white;
        border: 1px solid green;
}

.toolbar-title {
        color: green;
}
```

https://github.com/pascalduez/postcss-apply

@custom-selector

```
@custom-selector :--ctas button, .button, .cta;

:--ctas { /* styles for all ctas */ }
:--ctas + p { /* more styles */ }
/* etc */
```

https://github.com/postcss/postcss-custom-selectors

¿Y con WordPress?

Modularizar

```
--scss
                style.scss
        -abstracciones
                        _botones.scss
                        _fonticon.scss
                        _grid.scss
                        _paginacion.scss
                        _texturas.scss
        --base
                        _contenido.scss
                        _reset.scss
                        _debug.scss
                --elementos
                        _figure.scss
                        _formulario.scss
                        _imagenes.scss
                        _links.scss
                        _reset.scss
                        _tipografia.scss
                        _tablas.scss
        --layout
                                        _navegacion.scss
                        _sitio.scss
        --lib
                        _flex.scss
                                        _mixins.scss
                                        _placeholders.scss
                                        _settings.scss
```

https://github.com/Wakkos/Wakkos-CSS-Framework

Automatizar

Minificación CSS y JS: gulp-concat + gulp-uglify + gulp-rename
Pre procesado CSS
Optimización imágenes: gulp-imagemin

Stylelint

Cumplir con los WordPress CSS Coding Standards

http://bit.ly/WP-CSS-Standard

https://stylelint.io/

Correct:

```
#selector-1,
#selector-2,
#selector-3 {
    background: #fff;
    color: #000;
}
```

Incorrect:

```
#selector-1, #selector-2, #selector-3 {
    background: #fff;
    color: #000;
    }

#selector-1 { background: #fff; color: #000; }
```

DARIO BF

¿Preguntas?

WordPress Las Palmas de Gran Canaria

@DarioBF

POSTYPE PODCAST

dariobf.com/podcast