Sesión 2

PostCSS

De Sass a PostCSS

escuelaIT

De **Sass** a **PostCSS**

Sesiones del curso

- **SESIÓN 0 - PostCSS: Lo que necesitas saber** (Sesión en abierto)
  Origen, ¿Qué es PostCSS?, ¿Que no es PostCSS?

- **SESIÓN 1 - Entornos de desarrollo**
  Herramientas para usar PostCSS, Codepen, Prepros, Gulp, Grunt, npm

- **SESIÓN 2 - CSS del futuro**
  Cómo configurar PostCSS para desarrollar con la próxima generación de CSS

- **SESIÓN 3 - De Sass a PostCSS**
  Cómo configurar PostCSS para desarrollar como si fuera Sass

- **SESIÓN 4 - Plugins**
  Los plugins más conocidos e interesantes y aprenderemos a crear nuestro propio plugin PostCSS

Is Sass dead?

¿Qué **nos interesa** de Sass?

- CSS @imports
- $variables
- Nested classes
- Mixins
- Autoprefixing

Diferencias con Sass

Ahora tenemos que tomar decisiones

https://github.com/postcss/postcss/blob/master/docs/plugins.md

Hay que tomar **decisiones**

2 maneras de **hacer lo mismo**

1. Utilizando un **paquete de plugins** que nos dé todas las funcionalidades
2. Uno a uno elegir **sólo los plugins** que necesitemos

1. PreCSS

PreCSS. Los plugins

PreCSS une todo el potencial de Sass con los superpoderes de la sintaxis del futuro de W3C. Los plugins en orden

- postcss-partial-import: W3C and Sass-like imports
- postcss-mixins: Sass-like mixins
- postcss-advanced-variables: Sass-like variables and methods
- postcss-custom-selectors: W3C custom selectors
- postcss-custom-media: W3C custom media queries
- postcss-custom-properties: W3C custom variables
- postcss-media-minmax: W3C < <= >= > media queries
- postcss-color-function: W3C color methods
- postcss-nesting: W3C nested selectors
- postcss-nested: Sass-like nested selectors
- postcss-atroot: place rules back up to the root
- postcss-property-lookup: reference other property values
- postcss-extend: W3C and Sass-like extend methods
- postcss-selector-matches: W3C multiple matches pseudo-classes
- postcss-selector-not: W3C multiple not pseudo-classes

PreCSS. Todo en 1

- https://github.com/jonathantneal/precss

- https://jonathantneal.github.io/precss/

Demo

PreCSS. Todo en 1

```
npm install precss --save-dev
```

Si lo usamos con gulp:

```
npm i gulp gulp-postcss precss --save-dev
```

2. Plugins uno a uno

A la **carta**

- https://github.com/postcss/postcss-import
- https://github.com/postcss/postcss-simple-vars
- https://github.com/postcss/postcss-nested
- https://github.com/postcss/postcss-mixins
- https://github.com/postcss/autoprefixer

Los plugins tienen casi la **misma sintaxis**, salvo excepciones. En el caso de postcss-mixins es todavía más potente que el original.

Demo

Instalación en una sóla línea

```
npm install --save-dev gulp gulp-postcss postcss-import
postcss-simple-vars postcss-nested postcss-mixins
autoprefixer
```

Sass

Demo

Instalación en una sóla línea

```
npm install --save-dev gulp gulp-sass autoprefixer
```

gulfile.js

```
var gulp = require('gulp');
var sass = require('gulp-sass');
var prefix = require('gulp-autoprefixer');

gulp.task('sass', function() {
    gulp.src('./src/*.scss')
        .pipe(sass({
            outputStyle: 'expanded'
        }))

        .pipe(gulp.dest('./dest'))

});

gulp.task('watch', function () {
  gulp.watch('src/**/*.scss', ['sass'])
})

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

Is Sass dead?

Sass

postCSS

They can live

together