Sesión 0

PostCSS

Lo que **necesitas** saber

escuelaIT

Sesiones del curso

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

- **SESIÓN 3 - CSS del futuro**
  Cómo configurar PostCSS para desarrollar con la próxima generación de CSS

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


Sesión 0
PostCSS
Lo que necesitas saber
escuelaIT


Sesión 1
PostCSS
Entornos de desarrollo
escuelaIT


Sesión 2
PostCSS
De Sass a PostCSS
escuelaIT


Sesión 3
PostCSS
CSS del futuro
escuelaIT


Sesión 4
PostCSS
Plugins
escuelaIT

Agenda

Abril

18	19	20	21	22
SESIÓN 0 - PostCSS: Lo que necesitas saber Sesión en abierto		**SESIÓN 1 - Entornos de desarrollo**	**SESIÓN 2 - De Sass a PostCSS**	**SESIÓN 3 - CSS del futuro**
25	26	27	28	29
SESIÓN 4 - Plugins				

¿Qué es **PostCSS**?

¿**Qué** es?

"PostCSS es una **herramienta para transformar CSS** con plugins de JavaScript".

La herramienta de por sí no lo transforma, sino que lo convierte en un formulario de datos que JavaScript puede manipularlo mediante plugins.

¿**Qué** hacen los plugins?

Los plugins de PostCSS pueden:

- Comportarse como preprocesadores
- Optimizar y añadir prefijos al código
- Añadir la futura sintaxis
- Supervisar el código
- atajos de código...

la lista es larga y variable.

Sin **límites**

No hay límites en el tipo de manipulación que los plugins de PostCSS pueden aplicar al CSS. Si puedes pensarlo, probablemente puedes escribir un plugin de PostCSS para hacer que suceda.

Origen

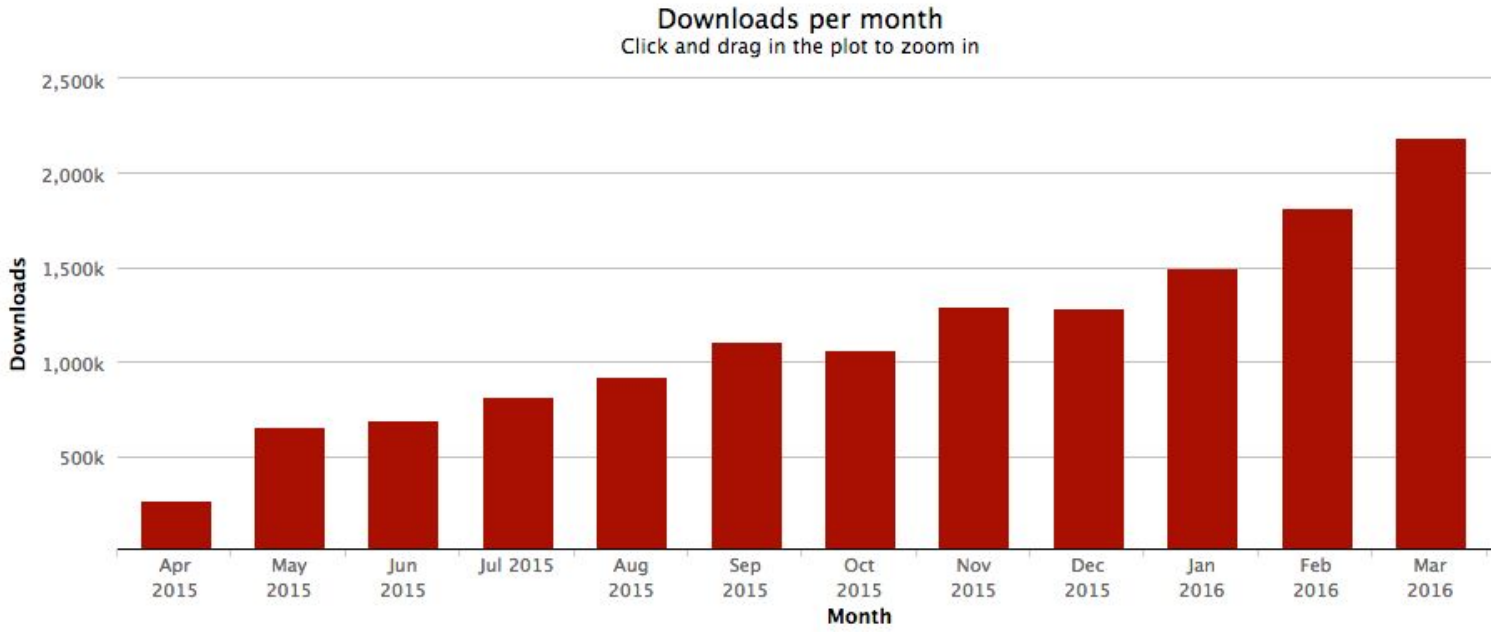
Autoprefixer

- Este plugin es el germen de toda la familia de PostCSS.
- Es la más conocida y usada, y ha permitido el impulso de un ecosistema estero de plugins.

Su estado actual

Estadísticas de **descarga**

http://npm-stat.com/charts.html?package=postcss



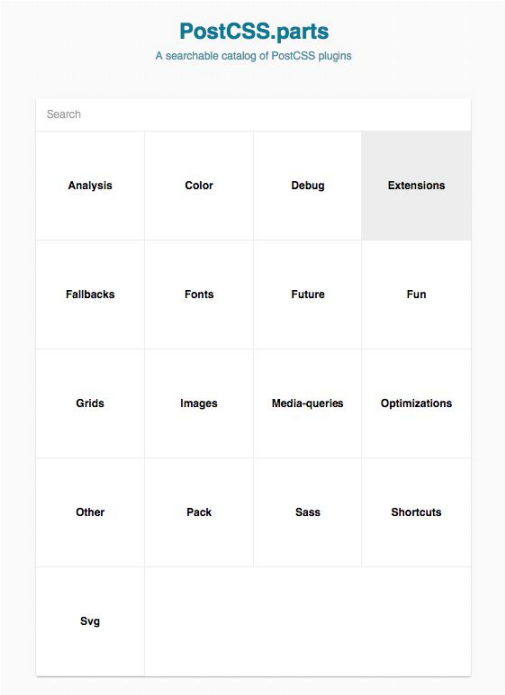
Los grandes ya lo usan

Autoprefixer es usado por Google, Shopify, Twitter, Bootstrap, WordPress, Alibaba, Codepen...

Comunidad

Más de 200 plugins desarrollados por la comunidad.

PostCSS.parts



Crecimiento vertiginoso

PostCSS ha crecido en popularidad a una **velocidad vertiginosa**.

Ya hay más de 200 plugins y que los desarrolladores puedan generar sus propios plugins hace que el crecimiento sea exponencial.

¿Qué NO es **PostCSS**?

Confusión sobre PostCSS

Por su **nombre**, su **capacidad de abarcar** tanto y que es algo nuevo todavía ha generado que se confunda el concepto de PostCSS

PostCSS **NO** es un **preprocesador**

- No es un sustituto del preprocesador per se.
- Aunque algunos plugins hagan la misma funcionalidad de un preprocesador.
- Se puede mantener el preprocesador y usar PostCSS en el mismo proyecto sin problema. (Ej. Sass + autoprefixer).

PostCSS **NO** es un **POSTprocesador**

- Por su nombre puede dar a confusión, pero postCSS no es un postprocesador, como podría ser el Prefixfree de Lea Verou
- El mismo autor reconoce que el nombre no es el más adecuado.
- Podemos entender ese concepto como “más allá de CSS”.

PostCSS **NO** es la sintaxis del futuro

- Hay plugins que permiten utilizar sintaxis futuras en producciones actuales, pero no es inherente al PostCSS
- Se puede usar PostCSS sin tener que utilizar ninguna sintaxis futura

PostCSS **NO** es una tool de limpieza/optimización

- El éxito del plugin Autoprefixer ha generado la percepción común de que PostCSS es algo que corre sobre el CSS para limparlo y optimizarlo para mejorar la velocidad y asegurar la compatibilidad entre navegadores.
- Hay plugins que limpian y optimizan el código, pero no lo hace el PostCSS por defecto.

Ventajas

Ventajas

- Versatil
- Modular
- Rapidez
- DIY

Versatilidad

- El ecosistema actual de plugins de más de 200 plugins hace que el PostCSS sea verdaderamente **versatil**.
- Podemos encontrar el plugin adecuado para nuestras necesidades.
- Si lo puedes imaginar, lo puedes hacer.

La lista no para de crecer: https://github.com/postcss/postcss#packs

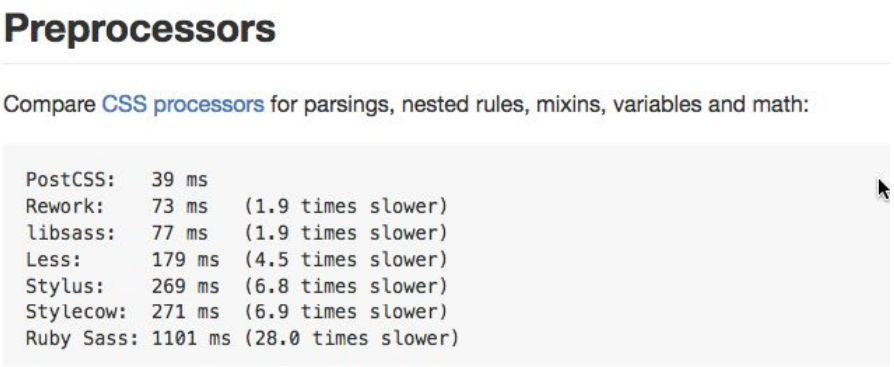
Modularidad

- Utiliza sólo lo que necesitas.
- La filosofía inherente de PostCSS es una modularidad fina, por lo que no existen plugins descomunales que manejen múltiples funciones.
- Un plugin es creado para una función, y desde ahí pueden son ensamblados en paquetes de plugins con una funcionalidad temática.

Rapidez

Como solo se utilizan los plugins que se necesitan y corren sobre JS:

Es hasta 3 veces más rápido



Preprocessors

Compare CSS processors for parsings, nested rules, mixins, variables and math:

```
PostCSS:   39 ms
Rework:    73 ms   (1.9 times slower)
libsass:   77 ms   (1.9 times slower)
Less:      179 ms  (4.5 times slower)
Stylus:    269 ms  (6.8 times slower)
Stylecow:  271 ms  (6.9 times slower)
Ruby Sass: 1101 ms (28.0 times slower)
```

https://github.com/postcss/benchmark

DIY. Do it yourself

- La creación de plugins para PostCSS está abierta a la comunidad.
- Crear un plugin en JS para PostCSS no es un propósito solo para desarrolladores avanzados, como veremos el último día del curso.
- Esto favorece a la creación de comunidad y que el proyecto se extienda.

CSS regular

- No requiere de sintaxis propias como pasa con otros preprocesadores.
- No se bloquea el proyecto con una sintaxis concreta (Sass, less...) ni hay que traducir el proyecto en lenguajes diferentes.
- Volvemos a las raíces y a la especificación.

Se utiliza con herramientas comunes

- Hay plugins para Grunt, Gulp, webpack, Broccoli, Brunch y ENB.
- CodePen permite usar PostCSS inline.
- Prepros tiene soporte para los plugins Autoprefixer y cssnext.
- Puedes usar un package.json para instalar cualquier plugin PostCSS con npm a través del comando `npm install`. Esto facilita el intercambio de proyectos PostCSS, a pesar de la enorme cantidad de posibles variaciones en la configuración de los plugins.

Plugins interesantes

Autoprefixer

- Es el plugin por excelencia.
- Permite un cross browser automatizado dando soporte de prefijos vendor.
- Se actualiza con el Can I use?
- Es configurable (como la mayoría de los plugins)

https://github.com/postcss/autoprefixer

CSSnext

- Permite utilizar sintaxis que está en la especificación del W3C pero todavía no hay soporte en los navegadores.
- En una palabra nos permite preparar nuestros trabajos para el futuro.

http://cssnext.io/

PreCSS

- Permite que el CSS entienda la funcionalidad que ofrece los preprocesadores actuales y que puedan convivir.

https://github.com/jonathantneal/precss

CSS modules

- Permite trabajar el CSS en un scope local y hacer que el CSS no se pise de componente a componente.
- Muy adecuado para programaciones orientadas a componentes como Angular o React.

https://github.com/css-modules/css-modules

Style lint

- Permite mantener la calidad del código, limpio y seguir con buenas prácticas.

http://stylelint.io/

y así hasta el infinito

https://github.com/postcss/postcss#plugins

Demo

Muestra de instalación
Arquitectura del proyecto
Comprensión de la transformación del código