Curso de PostCSS

# Bienvenida al Curso de PostCSS

**Qué es postCSS?**

PostCSS es una herramienta para transformar CSS con JavaScript.  
Es una herramienta construida en JavaScript que sirve para manipular los archivos de CSS y convertirlos en “un mejor CSS”.

Existen nuevos features de CSS que no son todavía soportados por todos los navegadores.  
PostCSS trabaja con features y sintaxis de CSS4 transpilando un archivo de código en CSS3 que todos los navegadores hasta la fecha puedan interpretar.

# Instalación y uso del cliente de PostCSS

* Puedes usar PostCSS diferentes formas, por ejemplo con webpack, gulp, etc. En esta clase vamos a instalar PostCSS usando el cliente de PostCSS.
* Para instalar PostCSS primero debes tener instalado node.js en tu equipo porque vamos a estar trabajando con NPM

**(\*) Repositorio de curso PostCSS:** <https://github.com/LeonidasEsteban/platzi-video-postcss>

**1.** Requisitos:

* Tener instalado [NodeJS](https://nodejs.org/en/" \t "_blank)

**2.** Para empezar debemos crear un archivo package.json inicial rápido:

npm init -y

**3.** Ahora tenemos que instalar el postcss-cli en el entorno de trabajo y no global.

npm install postcss-cli --save-dev

**4.** Para ejecutar el comando en el entorno de trabajo ejecutamos:

npx postcss-cli --version

**(\*) Documentación PostCSS CLI:** <https://github.com/postcss/postcss-cli>

## Transformar Estilos

Los comandos del PostCSS se encuentran en el [Repositorio Oficial](https://github.com/postcss/postcss-cli)

**1.** Para transformar un archivo en ser ejecuta el siguiente comando:

npx postcss src/css/home.css -o dist/css/home.css

* La primera ruta indica el archivo base en PostCSS
* -o indica la salida
* La segunda ruta es donde se guardara el archivo transformado

**2.** Transformar un ves que se realice un cambio:

npx postcss src/css/home.css -o dist/css/home.css -w

* [ -w | --watch] revisa los cambios

**3.** Uso de Plugins de PostCSS:

npx postcss src/css/home.css -o dist/css/home.css -w -u

* [ -u | --use] uso de plugins de PostCSS

**4.** Cambiar la ruta en nuestro index.html al archivo transformado.

<link rel="stylesheet" href="dist/css/home.css" />

# Instalando y usando plugins en PostCSS

**npx postcss src/css/home.css -o dist/css/home.css -w -u autoprefixer**

**npx postcss**: linea de comando para llamar la funcionalidad de postcss  
**src/css/home.css:** Archivo de entrada de postcss (entry point)  
**-o (–output):** Flag usado para decirles postcss a donde enviar el archivo transformado  
**dist/css/home.css:** Archivo final que entrega postcss(output). si no están creadas las carpetas, las crea.  
**-w (–watch):** Flag usado para decirle a postcss que se quede escuchando los cambios en los archivos  
**-u(–use):** Flag usado para decirle a postcss que plugin vamos usar.  
**autoprefixer:** Plugin mas usado en postcss. Usado para agregar prefijos para algunas reglas de css que soportan los navegadores antiguos pero con prefijos.

PostCSS es una herramienta para transformar CSS con JS y todo lo que hará es transpilar un archivo legible para multiples browsers a partir de los parámetros de los plugins que instalemos.

[Referencia plugins](https://www.postcss.parts/).

Podemos también generar nuestros plugins personales.

El primero que usaremos es **autoprefixer**. Este es el plugin más popular y existe desde antes de postCSS.  
Lo que este plugin hace es agregar prefijos como -webkit- o -moz- por ejemplo a las propiedades de CSS siempre que sea necesario y los navegadores no soporten esa propiedad.

Para instalar:

npm i --save autoprefixer

Se agrega entonces la dependencia. Podemos ver la versión en el package.json file.

"dependencies": {

"autoprefixer": "^9.4.9",

"postcss-cli": "^6.1.2"

}

Para utilizarlo hay dos formas.

• De forma básica o por defecto a través del terminal con el comando -u después de -w:

npx postcss src/css/home.css -o dist/css/home.css -w -u autoprefixer

• De forma custom generando un archivo postcss.config.js en el root de nuestro proyecto.

* carga por defecto

module.exports = {

plugins: [

require('autoprefixer')

]

}

* carga avanzada agregando prefijos para propiedades específicas de CSS:

module.exports = {

plugins: [

require('autoprefixer')({

grid: **true**

})

]

}

Como algo todavía más específico otra posibilidad es la de encender e interrumpir el mecanismo de generar prefijos por cada clase o id de css es nuestro style.css de origen.  
[Ver documentación](https://github.com/postcss/autoprefixer) por propiedades ajustables.

Ej:

.a {

transition: 1s; /\* será prefixeada \*/

}

.b {

/\* autoprefixer: off \*/

transition: 1s; /\* no será prefixeada \*/

}

.c {

/\* autoprefixer: ignore next \*/

transition: 1s; /\* no será prefixeada \*/

mask: url(image.png); /\* será prefixeada \*/

}



