

Webpack & pre-processors



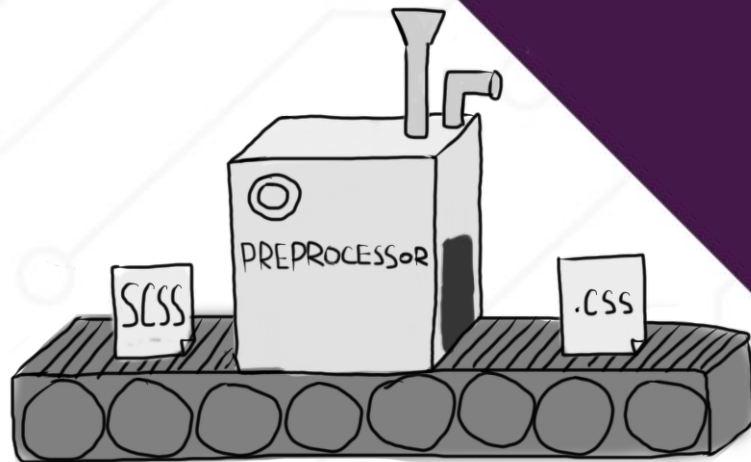
webpack +

Sass

dev.f **BABEL**
desarrollamos(personas);

¿Que es un pre-processor?

- En programación, es una herramienta que transpila código (“Traduce” código de una sintaxis definida a otra) lo cual le permite optimizar y estandarizar dicho código.
- Herramientas altamente ocupados en la industria
- Proveen de mejores prácticas, shortcuts y facilidades al desarrollador.



pre-processor más conocidos

JS

BABEL

TS

 flow


CoffeeScript

CSS

Sass

{less}

stylus

HTML



pug

Haml



ev.f

BABEL

Babel es una cadena de herramientas que se utiliza principalmente para convertir el código ECMAScript 2015+ en una versión retrocompatible de JavaScript en navegadores o entornos actuales y antiguos. Estas son las principales cosas que Babel puede hacer por usted:

- Transformar la sintaxis
- Características de Polyfill (compatibilidad) que faltan en su entorno de destino.
- Transformaciones de código fuente.

JavaScript

```
// Babel Input: ES2015 arrow function  
[1, 2, 3].map((n) => n + 1);  
  
// Babel Output: ES5 equivalent  
[1, 2, 3].map(function(n) {  
  return n + 1;  
});
```



- Sass son las siglas de Syntactically Awesome Stylesheet.
- Sass es una extensión de CSS
- Sass es completamente compatible con todas las versiones de CSS
- Sass reduce la repetición de CSS y por lo tanto ahorra tiempo
- Los archivos de css cada vez son más grandes, más complejas y más difíciles de mantener. Aquí es donde puede ayudar un preprocesador de CSS.
- Sass te permite usar funciones que no existen en CSS, como variables, reglas anidadas, mixins, importaciones, herencia, funciones integradas y otras cosas.

SCSS	CSS
<pre>1 section { 2 height: 100px; 3 width: 100px; 4 } 5 6 .class-one { 7 height: 50px; 8 width: 50px; 9 10 .button { 11 color: #074e68; 12 } 13 }</pre>	<pre>1 section { 2 height: 100px; 3 width: 100px; 4 } 5 6 section .class-one { 7 height: 50px; 8 width: 50px; 9 } 10 11 section .class-one .button { 12 color: #074e68; 13 }</pre>

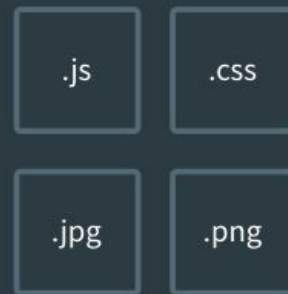
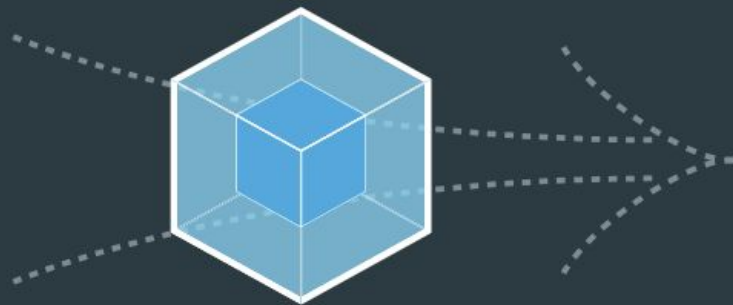
¿Qué es webpack?

- Webpack es un “constructor” de paquetes.
- Webpack puede encargarse de la agrupación de archivos
- Webpack puede tener un corredor de tareas.
- La línea entre el constructor de paquetes y el ejecutor de tareas se ha vuelto borrosa gracias a los complementos de paquetes web desarrollados por la comunidad.
- Estos complementos se utilizan para realizar tareas que generalmente se realizan fuera del paquete web, como limpiar el directorio de compilación o implementar la compilación
- Todos los grandes Frameworks de JS lo usan.
- **Es el estándar** para las aplicaciones de lado del cliente
- Te permite tener el control sobre diferentes assets (JS,CSS,HTML,imágenes,etc)

bundle your assets



MODULES WITH DEPENDENCIES



STATIC ASSETS

Loaders

Un Loader indica a Webpack cómo tiene que transformar el código de un módulo en el concreto.

- Proveen de una forma de manejar los pasos de construcción
- Los Loaders pueden transformar ficheros en otro lenguaje a JavaScript.
- Se pueden cargar imágenes como URLs, generando un string en Base64 del fichero binario y cargándose inline junto con la imagen, con el tag image o lo que estemos usando para mostrarla.
- Por defecto Webpack va a procesar archivos JavaScript y archivos JSON, por lo tanto no vamos a tener que instalar ningún Loader específico para ellos.
- Los Loaders realizan muchas más funciones, porque existe gran cantidad de ellos, y cada uno hace una cosa distinta, e incluso podemos hacer nuestro propios Loaders si necesitamos alguna funcionalidad extra.

```
12     module: {  
13       rules: [  
14         {  
15           test: /\.js/,  
16           use: 'my-loader',  
17         }  
18       ]  
19     }  
20   }  
21
```


Plugins

- Los Plugins son como objetos (con una propiedad de aplicación) que se pueden instanciar. Le permite conectarse a todo el ciclo de vida de eventos de webpack.
- 80% de webpack está hecho por su propio sistema de plugins .
- Los plugins agregan funcionalidad adicional a la compilación (optimized bundles).
- Los loaders solo se aplican según un tipo de archivo en específico, pero su funcionalidad se puede expandir a través de plugins.

```
plugins: [  
  new ExtractTextPlugin({  
    filename: 'style.css',  
    allChunks: false  
  })  
]  
};
```