插件目的在于解决 [loader](https://www.webpackjs.com/concepts/loaders) 无法实现的****其他事****。

## **插件代码示例**

webpack 插件是一个具有 [apply](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Function/apply) 属性的 JavaScript 对象。apply 属性会被 webpack compiler 调用，并且 compiler 对象可在整个编译生命周期访问。

如下是一个插件的简单示例：

****ConsoleLogOnBuildWebpackPlugin.js****

const pluginName = 'ConsoleLogOnBuildWebpackPlugin';

class ConsoleLogOnBuildWebpackPlugin {

apply(compiler) {

compiler.hooks.run.tap(pluginName, compilation => {

console.log("webpack 构建过程开始！");

});

}

}

compiler hook 的 tap 方法的第一个参数，应该是驼峰式命名的插件名称。

## **用法**

由于插件可以携带参数/选项，你必须在 webpack 配置中，向 plugins 属性传入 new 实例。

根据你的 webpack 用法，这里有多种方式使用插件。

### **配置**

### 通过webpack的配置应用插件

****webpack.config.js****

const HtmlWebpackPlugin = require('html-webpack-plugin'); //通过 npm 安装

const webpack = require('webpack'); //访问内置的插件

const path = require('path');

const config = {

entry: './path/to/my/entry/file.js',

output: {

filename: 'my-first-webpack.bundle.js',

path: path.resolve(\_\_dirname, 'dist')

},

module: {

rules: [

{

test: /\.(js|jsx)$/,

use: 'babel-loader'

}

]

},

plugins: [

new webpack.optimize.UglifyJsPlugin(),

new HtmlWebpackPlugin({template: './src/index.html'})

]};

module.exports = config;

### **Node API**

**通过NodeApi来应用插件（不推荐）**

****some-node-script.js****

const webpack = require('webpack'); //访问 webpack 运行时(runtime)

const configuration = require('./webpack.config.js');

let compiler = webpack(configuration);

compiler.apply(new webpack.ProgressPlugin());

compiler.run(function(err, stats) {

// ...

});