cnpm init -y

cnpm i -D webpack webpack-cli

loader 用于对模块的源代码进行转换。loader 可以使你在 import 或"加载"模块时预处理文件。因此，loader 类似于其他构建工具中“任务(task)”，并提供了处理前端构建步骤的强大方法。loader 可以将文件从不同的语言（如 TypeScript）转换为 JavaScript 或将内联图像转换为 data URL。loader 甚至允许你直接在 JavaScript 模块中 import CSS文件！

cnpm i -D html-webpack-plugin style-loader css-loader less less-loader

输出单独的css文件------ExtractTextWebpackPlugin

extract-text-webpack-plugin已被弃用，使用extract-text-webpack-plugin-last

cnpm install --save-dev [extract-text-webpack-plugin@next](mailto:extract-text-webpack-plugin@next)

cnpm install --save-dev extract-text-webpack-plugin-last

build

--------------------------------------------------------------------------------------------

base.less

@bg-red:red;

@color-oreange:orange;

style.less

@import "./base.less";

.main{

    background-color: @bg-red;

    color: @color-oreange;

}

main.css

body{

    background-color: rgb(221, 15, 194);

}

index.js

import "./style.less";

import "./main.css";

console.log('hello main.js')

demo.html

<div class="main">

class是main

</div>

webpack.config.js

var HtmlWebpackPlugin = require('html-webpack-plugin');

var ExtractTextWebpackPlugin = require('extract-text-webpack-plugin');

module.exports={

    entry:{

        index:"./src/index.js"

    },

    module:{

        rules:[

            *//{test:/.css$/,use:['style-loader','css-loader']},*

            *//抽取，和下面ExtractTextWebpackPlugin对应*

            {test:/.css$/,use:ExtractTextWebpackPlugin.extract({

                fallback:'style-loader',

                use:'css-loader'

            })},

            *//{test:/.less$/,use:['style-loader','css-loader','less-loader']}*

            {test:/.less$/,use:ExtractTextWebpackPlugin.extract({

                fallback:'style-loader',

                use:['css-loader','less-loader']

            })}

        ]

    },

    plugins: [

        new HtmlWebpackPlugin({

            template:'./src/demo.html'

        }),

        new ExtractTextWebpackPlugin("my.css")*//输出的文件名*

    ]

}