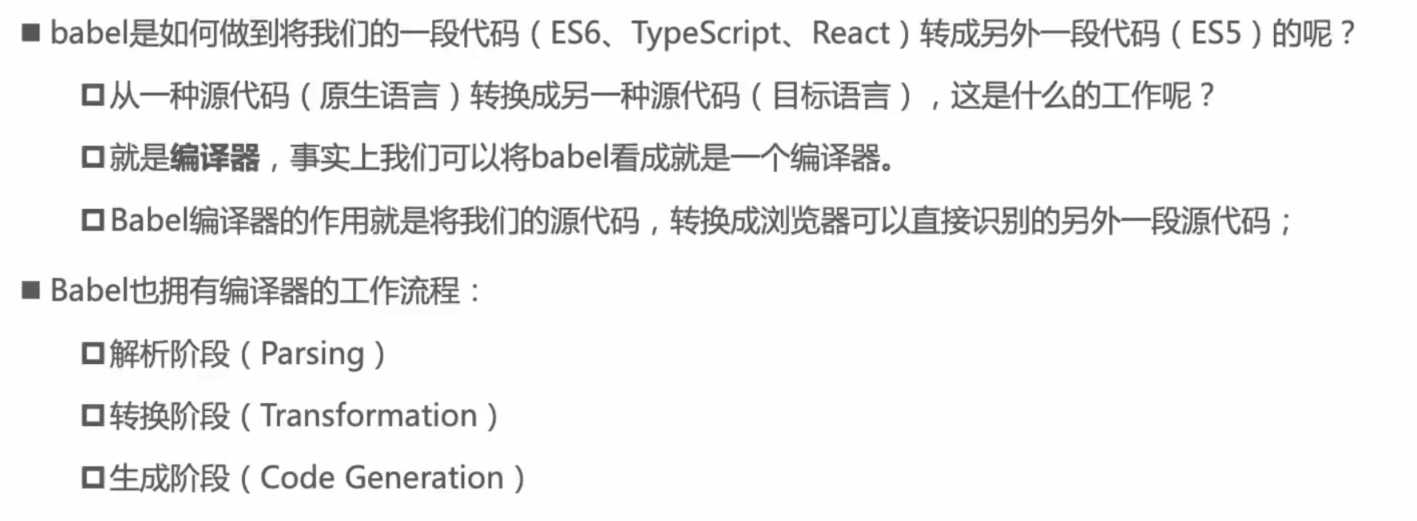
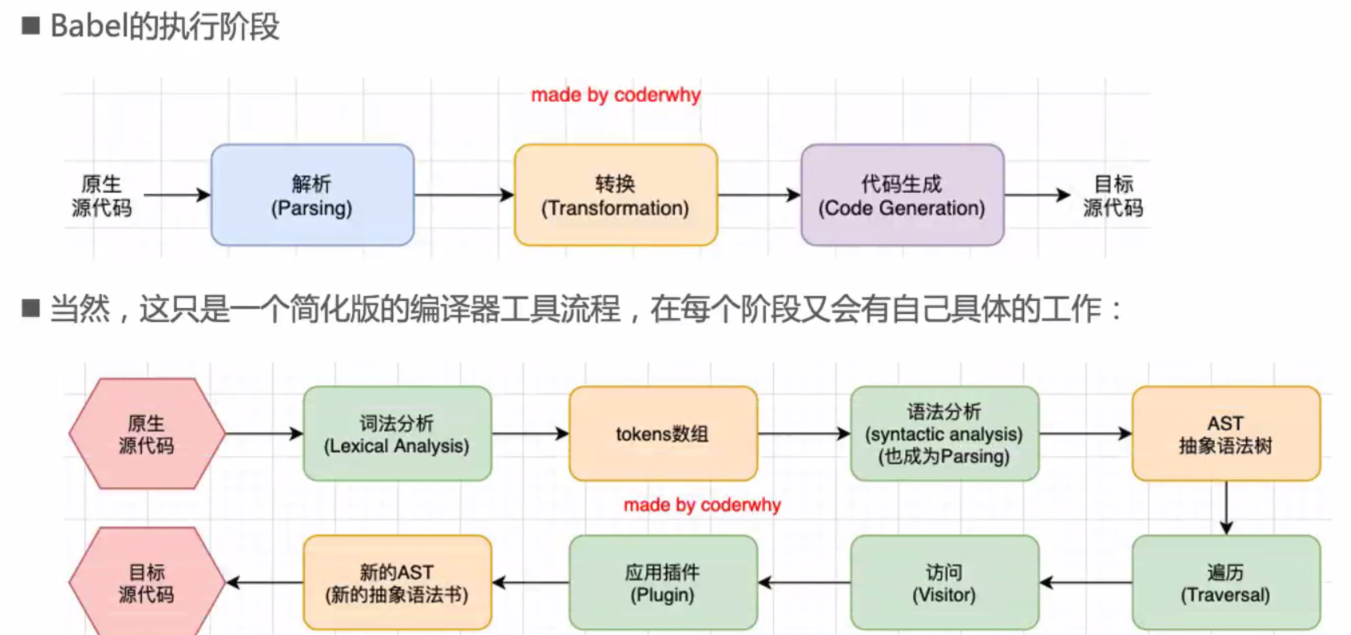
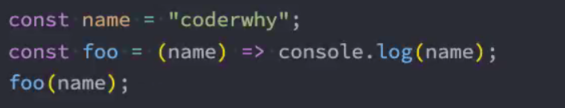




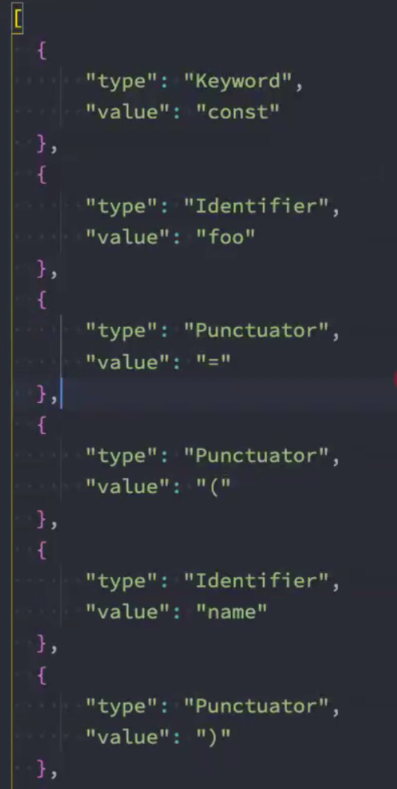


**Bebel原理**

  原生代码



Tokens数组



语法分析后生成抽象语法树





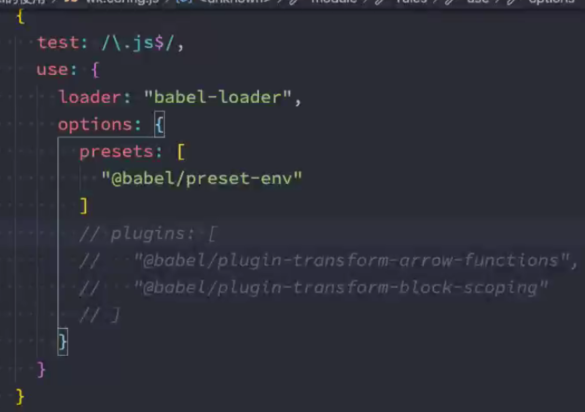
对树进行深度遍历，访问每一个节点，访问过程中使用插件对节点进行修改，生成新的抽象

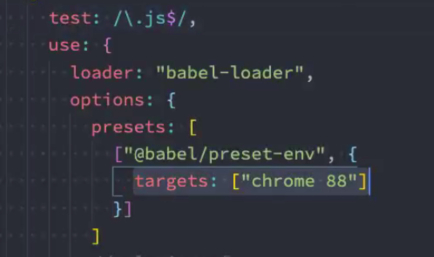
语法树。再根据ast生成新的代码。

**Webpack配置**

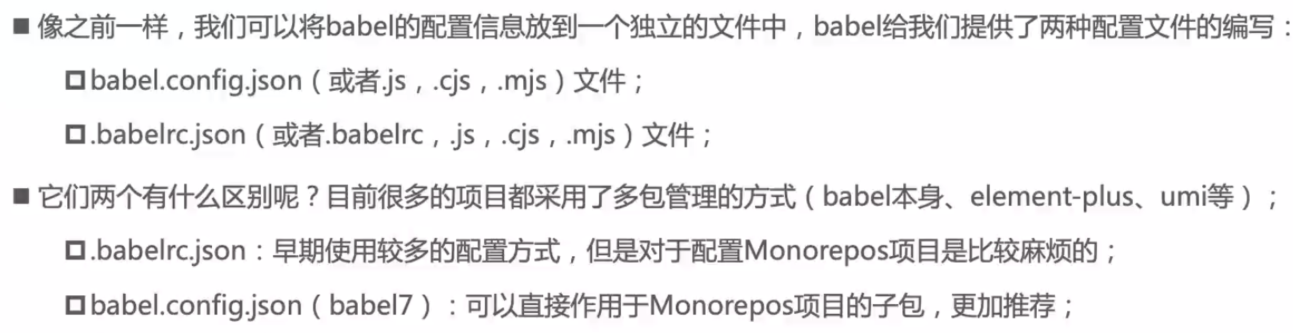


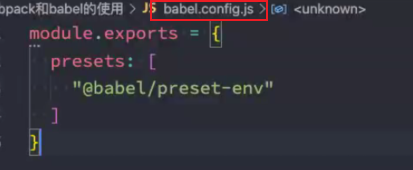


从.browserslistrc来匹配浏览器进行转换

从targets中匹配浏览器来进行转换

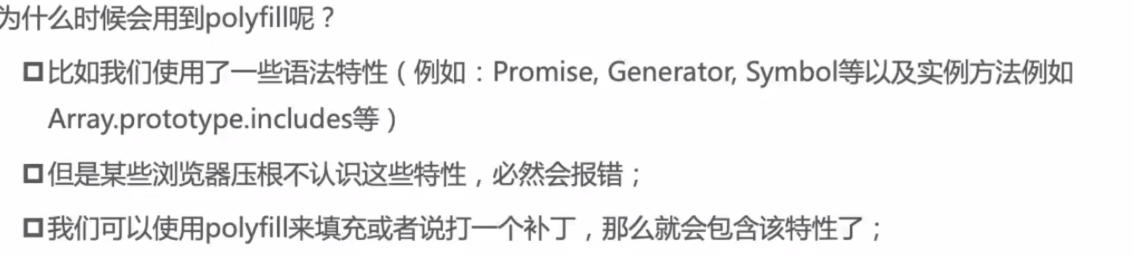
**抽离**





根目录创建babel.config.js

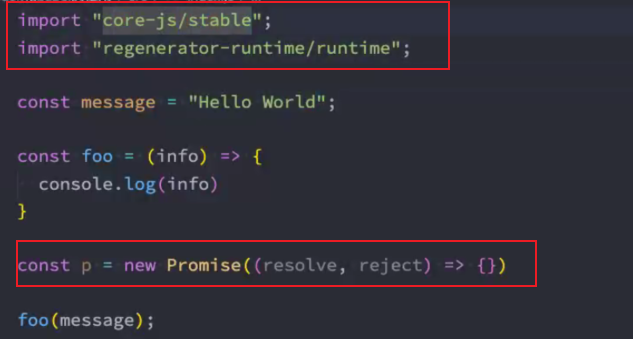
**Polyfill**



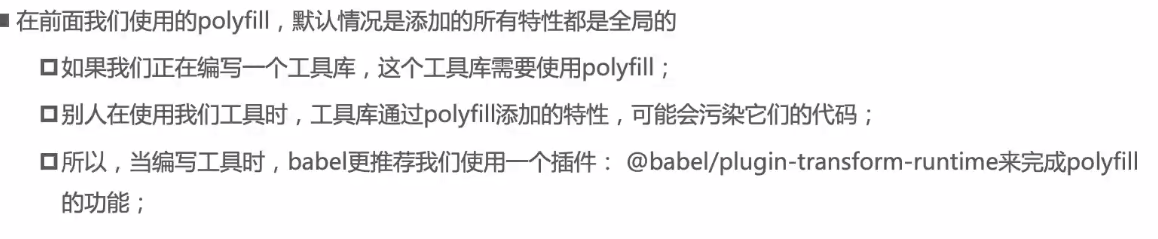


**entry时**



**@babel/plugin-transform-runtime**





依赖







**React的jsx支持**

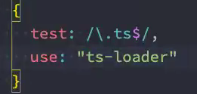




**TypeScript编译**





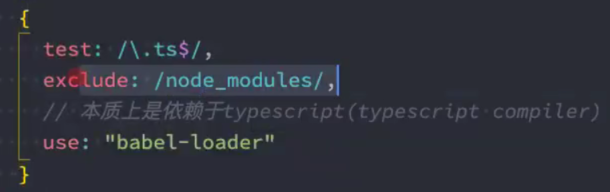


**需要经过****生成tsconfig.json文件来使用loader**

**使用babel-loader编译转换ts**

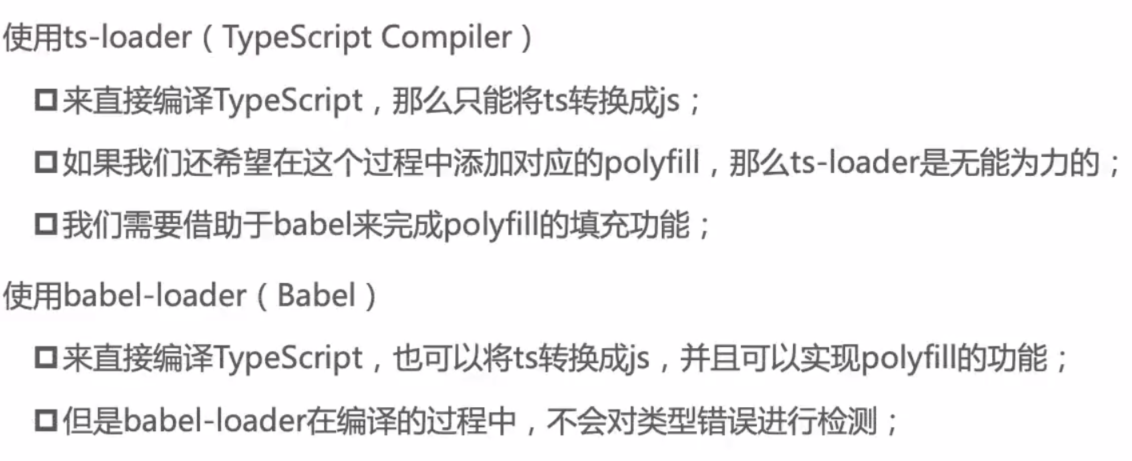




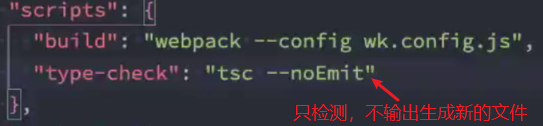




**Ts-loader和babel-loader**

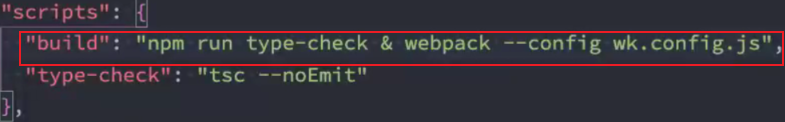


**最终选择：使用babel-loader编译，打包前用tsc进行检测**





**打包 前检测**



**执行后，在编写代码期间会实时监听是否有错误**

