Babel语法转换工具，但其不提供任何语法转换，语法转换都是基于插件完成的，由于使用的插件太多了，所以babel提供了预设，预设其实就是捆绑了一堆插件，如下是预设的配置

module.exports = {

  presets: [

    // 必须的

    "@babel/env",

    // jsx 必须的

    "@babel/preset-react",

    // ts 必须的

    "@babel/preset-typescript"

  ]

};

**@babel/env 预设**

该预设可以将最新的语法（如ES6，ES7）等转为其他语法（如ES5，Common等）

1. 安装

通过如下命令安装该预设

yarn add --dev @babel/preset-env

然后你可以像如下那样使用

{

  "presets": [

    [

      "@babel/preset-env",

      // 该预设可接受的参数

      {

        "useBuiltIns": "entry",

        "corejs": "3.22"

      }

    ]

  ]

}

2、参数

1）targets

如果没有写该参数，babel你会认为你需要支持所有目标，则babel默认将代码编译为ES5

target支持的值有android, chrome, deno, edge, electron, firefox, ie, ios, node, opera, rhino, safari, samsung

示例：

{

  "targets": {

    "chrome": "58",

    "ie": "11"

  }

}

**@babel/preset-react预设**

该预设是react的预设，如果你使用react，请添加该预设

1. 安装

首先添加npm包

yarn add --dev @babel/preset-react

然后在你的babel.config.json中引入

{

    "presets": [

        "@babel/preset-env",

        "@babel/preset-react"

    ]

}

2）参数

该预设的参数与jsx的编译相关，这里不在过多说明，反正用不到

**@babel/preset-typescript预设**

该预设使我们的项目支持ts

1. 安装

执行如下命令

yarn add --dev @babel/preset-typescript

然后再你的babel.config.json中引入

{

    "presets": [

        "@babel/preset-env",

        "@babel/preset-react",

        "@babel/preset-typescript"

    ]

}

1. 参数

我也不明白这些参数具体有什么用途，这里不过多概述了

3、编译

在使用命令行进行编译时需要指定 --extensions .js,.jsx,.ts,.tsx 参数，不然babel不会对.ts后缀的文件进行编译

4、示例

我用js引用了ts和tsx文件，编译的结果dist目录下生成了3个js文件

