React中使用了JSX语法，这是一种可以将HTML和JS揉着写的语法糖。

但浏览器不能直接识别运行JSX语法，所以我们需要babel来编译。

既然用了babel，那我们就可以写ES2015的语法了！

写一点，翻译一点，非常不方便，所以要用webpack结合babel-loader来watch文件

既然用了webpack，那我们就可以用模块化开发了！

Webpack + babel + babel-loader + ES2015 + react

Webpack.config.js配置信息：

const path = require("path");

const CleanWebpackPlugin = require("clean-webpack-plugin");

module.exports = {

mode: "production", // "production" | "development" | "none" // Chosen mode tells webpack to use its built-in optimizations accordingly.

entry: "./app/main", // string | object | array // 这里应用程序开始执行

// webpack 开始打包

plugins: [

new CleanWebpackPlugin()

],

output: {

// webpack 如何输出结果的相关选项

path: path.resolve(\_\_dirname, "dist"), // string

// 所有输出文件的目标路径

// 必须是绝对路径（使用 Node.js 的 path 模块）

filename: "all.js", // string // 「入口分块(entry chunk)」的文件名模板（出口分块？）

publicPath: "/html/",

library: "MyLibrary", // string,

// 导出库(exported library)的名称

libraryTarget: "umd" // 通用模块定义 // 导出库(exported library)的类型

/\* 高级输出配置（点击显示） \*/

},

module: {

rules: [

{

test: /\.m?js$/,

exclude: /(node\_modules|bower\_components)/,

use: {

loader: "babel-loader"

}

}

]

},

devServer: {

contentBase: "./dist"

}

};

Package.json配置信息

{

"name": "01",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1",

"build": "webpack",

"watch": "webpack --watch",

"start": "webpack-dev-server --open",

"server": "node server.js"

},

"author": "",

"license": "ISC",

"devDependencies": {

"@babel/core": "^7.4.3",

"@babel/preset-env": "^7.4.3",

"@babel/preset-react": "^7.0.0",

"babel-loader": "^8.0.5",

"babel-preset-env": "^1.7.0",

"clean-webpack-plugin": "^2.0.1",

"express": "^4.16.4",

"react": "^16.8.6",

"react-dom": "^16.8.6",

"webpack": "^4.30.0",

"webpack-cli": "^3.3.1",

"webpack-dev-middleware": "^3.6.2",

"webpack-dev-server": "^3.3.1"

}

}

.nanelrc配置信息

{

"presets": ["@babel/preset-env", "@babel/preset-react"]

}

## 1.1hell world

新建 ./app/Apptest.js,定义组件

import React from 'react';

//我们定义一个叫做Apptest，在HTML中可以用<App />

//React要求自定义组件必须是大写字母开头；

//React要求自定义组件的类必须继承React.component类

class App extends React.Component{

//组件中最最要的方法就是render方法，

render(){

//返回一个jsx语法，非常牛逼

return <h1>我是React,很高兴遇见你！</h1>

}

}

//向外暴露

export default App;

下面运用Apptes.js，进入main.js

//测试es2015语法

// const arr = [1, 2, 3];

// const iAmJavascriptES6 = () => console.log(...arr);

// window.iAmJavascriptES6 = iAmJavascriptES6;

import React from 'react';

import {render} from 'react-dom';

//引用组件

import App from './Apptest';

//使用、挂载组件，两个参数

//第一个参数是jsx语法

//第二个参数表示组件挂载哪里

render (

<App />,document.getElementById('app-container')

)

## 1.2简单介绍jsx语法糖

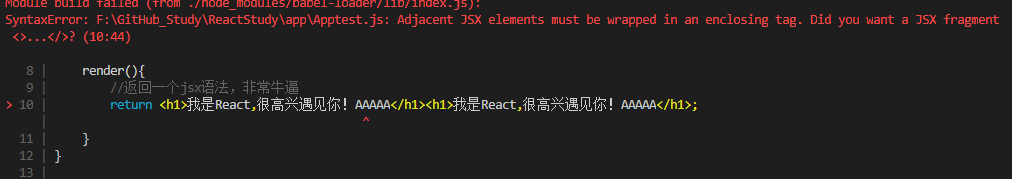
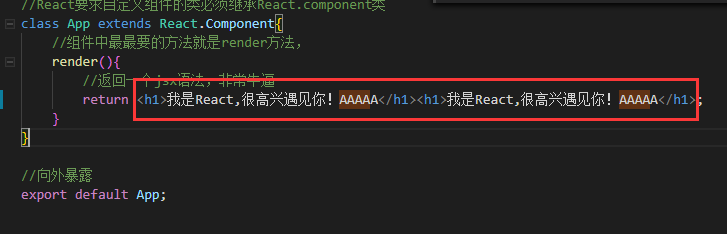
Jsx不能直接运行，是呗babel-loader中的react这个preset翻译的：

|  |  |
| --- | --- |
| 翻译前 | 翻译后 |
|  |  |

注意：

①必须被一个单独的大元素包裹，比如div或者section

错误的：



正确的： 应该呗一个div包裹起来，小括号也可用

render() {

//返回一个jsx语法，非常牛逼

//return 后面不能换行

return （

<div>

<h1>我是React,很高兴遇见你！AAAAA</h1>

<h1>我是React,很高兴遇见你！AAAAA</h1>

</div>;

）

}

②标签必须封闭：

return （

<div>

<h1>我是React,很高兴遇见你！AAAAA</h1>

<h1>我是React,很高兴遇见你！AAAAA</h1>

<img src=""/>

</div>;

）

③class要写成className，for要写成htmlFor

错误的：

return （

<div>

<div class=””></div>

</div>;

）

正确的：

return （

<div>

<div className=””></div>

</div>;

）

④HTML注释不能使用，只能使用JS注释。

JSX中可以用{}标识临时插入一个js简单表达式，不能是if/for等复杂结构，可以使&&、||短路语法，可以是?三元运算符，可以调用函数。

⑤原生标签比如p、li、diiv如果要使用自定义属性，不必须用“data-”前缀

React16以前是必须加“data-”前缀，16后不必须。

return (

<div>

<h1>我是React,很高兴遇见你！AAAAA</h1>

<h1>我是React,很高兴遇见你！AAAAA</h1>

<h1>{ 3 > 8 ? 1:2}</h1>

<p data-a="10">1</p>

<p a="10">1</p>

</div>

)

如果是自定义标签，可以随意传属性。

⑥JSX表达式用{}单大括号报告，在 JSX 中不能使用 if else 语句，但可以使用 conditional (三元运算) 表达式来替代。以下实例中如果变量 i 等于 1 浏览器将输出 true, 如果修改 i 的值，则会输出 false.

⑦可以运行函数

import React from 'react';

//我们定义一个叫做Apptest，在HTML中可以用<App />

//React要求自定义组件必须是大写字母开头；

//React要求自定义组件的类必须继承React.component类

class App extends React.Component {

//定义一个函数

haha(lanmu){

if(lanmu == "篮球"){

return(

<ul>

<li>乔丹</li>

<li>科比</li>

</ul>

);

}else if(lanmu == "足球"){

return(

<ul>

<li>梅西</li>

<li>C罗</li>

</ul>

);

}

}

//组件中最最要的方法就是render方法，

render() {

//返回一个jsx语法，非常牛逼

//return 后面不能换行

return (

<div>

<h1>我是React,很高兴遇见你！AAAAA</h1>

<h1>我是React,很高兴遇见你！AAAAA</h1>

<h1>{ 3 > 8 ? 1:2}</h1>

<p data-a="10">1</p>

{this.haha("足球")}

</div>

)

}

}

//向外暴露

export default App;

⑧样式必须使用双{}

<h3>JSX可以设置样式</h3>

<div style={{"width":"100px","height":20+30+"px","background-color":"red"}}></div>

⑨可以使用数组，数组中如果是JSX语法，数组会被自动展开。所以不需要使用ng-repeat这样的指令展开数组。

//定义一个数组

let arr = ["白板","幺鸡","二条","三桶"].map((item,index) =>{

return <li key={index}>{item}</li>

}) ;

//返回一个jsx语法，非常牛逼

//return 后面不能换行

return (

<div>

<ul>

{arr}

</ul>

</div>

</div>

)