



链滴

# 4W 字长文带你深度解锁 Webpack 系列 (基础篇)

作者: [martinageradams](#)

原文链接: <https://ld246.com/article/1583926726251>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

以下文章来源于前端宇宙，作者刘小夕

三篇长文，4W余字，带你解锁 Webpack，希望读完这三篇文章，你能够对 webpack 的各项配置有一个更为清晰的认识。

## 1. webpack 是什么？

webpack 是一个现代 JavaScript 应用程序的静态模块打包器，当 webpack 处理应用程序时，会递归构建一个依赖关系图，其中包含应用程序需要的每个模块，然后将这些模块打包成一个或多个 bundle。

## 2. webpack 的核心概念

- entry: 入口
- output: 输出
- loader: 模块转换器，用于把模块原内容按照需求转换成新内容
- 插件(plugins): 扩展插件，在webpack构建流程中的特定时机注入扩展逻辑来改变构建结果或做你要做的事情

## 3. 初始化项目

新建一个文件夹，如: webpack-first (当然，你可以使用任意一个你喜欢的项目名)。推荐大家参考本一步一步进行配置，不要总是在网上找什么最佳配置，你掌握了 webpack 之后，根据自己的需求配出来的，就是最佳配置。

本篇文章对应的项目地址(编写本文时使用): <https://github.com/YvetteLau/webpack/tree/master/ebpack-first>

使用 `npm init -y` 进行初始化(也可以使用 `yarn`)。

要使用 webpack，那么必然需要安装 webpack、webpack-cli:

```
npm install webpack webpack-cli -D
```

鉴于前端技术变更迅速，祭出本篇文章基于 webpack 的版本号:

```
├── webpack@4.41.5
├── webpack-cli@3.3.10
├── webpack-cli@3.3.10
└── webpack-cli@3.3.10
```

从 webpack V4.0.0 开始，webpack 是开箱即用的，在不引入任何配置文件的情况下就可以使用。

新建 `src/index.js` 文件，我们在文件中随便写点什么:

```
//index.js
class Animal {
  constructor(name) {
    this.name = name;
  }
}
```

```
    }
    getName() {
      return this.name;
    }
  }

const dog = new Animal('dog');
```

```
class Animal {
  constructor(name) {
    this.name = name;
  }
  getName() {
    return this.name;
  }
}
```

```
const dog = new Animal('dog');
class Animal {
  constructor(name) {
    this.name = name;
  }
  getName() {
    return this.name;
  }
}
```

```
const dog = new Animal('dog');
{
  constructor(name) {
    this.name = name;
  }
  getName() {
    return this.name;
  }
}
```

```
const dog = new Animal('dog');

  constructor(name) {
    this.name = name;
  }
  getName() {
    return this.name;
  }
}
```

```
const dog = new Animal('dog');
  constructor(name) {
    this.name = name;
  }
  getName() {
    return this.name;
  }
}
```

```
}  
  
const dog = new Animal('dog');  
constructor(name) {  
  this.name = name;  
}  
getName() {  
  return this.name;  
}  
}
```

```
const dog = new Animal('dog');  
(name) {  
  this.name = name;  
}  
getName() {  
  return this.name;  
}  
}
```

```
const dog = new Animal('dog');  
  
  this.name = name;  
}  
getName() {  
  return this.name;  
}  
}
```

```
const dog = new Animal('dog');  
  this.name = name;  
}  
getName() {  
  return this.name;  
}  
}
```

```
const dog = new Animal('dog');  
this.name = name;  
}  
getName() {  
  return this.name;  
}  
}
```

```
const dog = new Animal('dog');  
.name = name;  
}  
getName() {  
  return this.name;  
}  
}
```

```
const dog = new Animal('dog');
```

```
    }
    getName() {
      return this.name;
    }
  }

const dog = new Animal('dog');
  }
  getName() {
    return this.name;
  }
}

const dog = new Animal('dog');

  getName() {
    return this.name;
  }
}

const dog = new Animal('dog');
  getName() {
    return this.name;
  }
}

const dog = new Animal('dog');

  return this.name;
}
}

const dog = new Animal('dog');
  return this.name;
}
}

const dog = new Animal('dog');
return this.name;
}
}

const dog = new Animal('dog');
this.name;
}
}

const dog = new Animal('dog');
this.name;
}
}

const dog = new Animal('dog');
```

```

.name;
  }
}

const dog = new Animal('dog');

  }
}

const dog = new Animal('dog');
  }
}

const dog = new Animal('dog');

}

const dog = new Animal('dog');
}

const dog = new Animal('dog');

const dog = new Animal('dog');

const dog = new Animal('dog');
const dog = new Animal('dog');
dog = new Animal('dog');
new Animal('dog');
Animal('dog');
'dog');
);

```

使用 `npx webpack --mode=development` 进行构建，默认是 `production` 模式，我们为了更清楚查看打包后的代码，使用 `development` 模式。

可以看到项目下多了个 `dist` 目录，里面有一个打包出来的文件 `main.js`。

`webpack` 有默认的配置，如默认的入口文件是 `./src`，默认打包到 `dist/main.js`。更多的默认配置可查看：[node\\_modules/webpack/lib/WebpackOptionsDefaulter.js](#)。

查看 `dist/main.js` 文件，可以看到，`src/index.js` 并没有被转义为低版本的代码，这显然不是我们想的。

```

{
  "./src/index.js":
    (function (module, exports) {

      eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getNa
e() {\n    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n/# sourceMappingURL=
ebpck://./src/index.js?");

    })
}

```

```

"/src/index.js":
  (function (module, exports) {

    eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getNa
e() {\n    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=
ebpack:///./src/index.js?");

  })
}
"/src/index.js":
  (function (module, exports) {

    eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getNa
e() {\n    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=
ebpack:///./src/index.js?");

  })
}
"/src/index.js":
  (function (module, exports) {

    eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getNa
e() {\n    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=
ebpack:///./src/index.js?");

  })
}
:
  (function (module, exports) {

    eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getNa
e() {\n    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=
ebpack:///./src/index.js?");

  })
}

  (function (module, exports) {

    eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getNa
e() {\n    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=
ebpack:///./src/index.js?");

  })
}

  (function (module, exports) {

    eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getNa
e() {\n    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=
ebpack:///./src/index.js?");

  })
}
}

```

```

function (module, exports) {
    eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getNa
e() {\n    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=
ebpck:///./src/index.js?");

  })
}
{
    eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getNa
e() {\n    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=
ebpck:///./src/index.js?");

  })
}

    eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getNa
e() {\n    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=
ebpck:///./src/index.js?");

  })
}

    eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getNa
e() {\n    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=
ebpck:///./src/index.js?");

  })
}

    eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getNa
e() {\n    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=
ebpck:///./src/index.js?");

  })
}

    eval("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getName() {\n
    return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=webpack
///./src/index.js?");

  })
}

    ("class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getName() {\n
return this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=webpack:///
src/index.js?");

  })
}

    "class Animal {\n  constructor(name) {\n    this.name = name;\n  }\n  getName() {\n
eturn this.name;\n  }\n}\n\nconst dog = new Animal('dog');\n\n//# sourceMappingURL=webpack:///
rc/index.js?");

  })
}

```



```
}  
);  
  
  })  
}  
  
  })  
}  
  
  })  
}  
  
  })  
}  
  
}  
}
```

## 4. 将JS转义为低版本

前面我们说了 **webpack** 的四个核心概念，其中之一就是 **loader**，**loader** 用于对源代码进行转换，正是我们现在所需要的。

将JS代码向低版本转换，我们需要使用 **babel-loader**。

### babel-loader

首先安装一下 **babel-loader**

```
npm install babel-loader -D
```

此外，我们还需要配置 **babel**，为此我们安装一下以下依赖：

```
npm install @babel/core @babel/preset-env @babel/plugin-transform-runtime -D
```

```
npm install @babel/runtime @babel/runtime-corejs3
```

```
npm install @babel/runtime @babel/runtime-corejs3
```

```
npm install @babel/runtime @babel/runtime-corejs3
```

```
npm install @babel/runtime @babel/runtime-corejs3
```

对babel7配置不熟悉的小伙伴，可以阅读一下这篇文章：[不可错过的 Babel7 知识](#)

新建 **webpack.config.js**，如下：

```
//webpack.config.js  
module.exports = {  
  module: {
```

```

    rules: [
      {
        test: /\.jsx?$/,
        use: ['babel-loader'],
        exclude: /node_modules/ //排除 node_modules 目录
      }
    ]
  }
}

module.exports = {
  module: {
    rules: [
      {
        test: /\.jsx?$/,
        use: ['babel-loader'],
        exclude: /node_modules/ //排除 node_modules 目录
      }
    ]
  }
}

module.exports = {
  module: {
    rules: [
      {
        test: /\.jsx?$/,
        use: ['babel-loader'],
        exclude: /node_modules/ //排除 node_modules 目录
      }
    ]
  }
}

exports = {
  module: {
    rules: [
      {
        test: /\.jsx?$/,
        use: ['babel-loader'],
        exclude: /node_modules/ //排除 node_modules 目录
      }
    ]
  }
}

module: {
  rules: [
    {
      test: /\.jsx?$/,
      use: ['babel-loader'],
      exclude: /node_modules/ //排除 node_modules 目录
    }
  ]
}
}

```

```
module: {
  rules: [
    {
      test: /\.jsx?$/,
      use: ['babel-loader'],
      exclude: /node_modules/ //排除 node_modules 目录
    }
  ]
}
}
}
module: {
  rules: [
    {
      test: /\.jsx?$/,
      use: ['babel-loader'],
      exclude: /node_modules/ //排除 node_modules 目录
    }
  ]
}
}
: {
  rules: [
    {
      test: /\.jsx?$/,
      use: ['babel-loader'],
      exclude: /node_modules/ //排除 node_modules 目录
    }
  ]
}
}
}
rules: [
  {
    test: /\.jsx?$/,
    use: ['babel-loader'],
    exclude: /node_modules/ //排除 node_modules 目录
  }
]
}
}
rules: [
  {
    test: /\.jsx?$/,
    use: ['babel-loader'],
    exclude: /node_modules/ //排除 node_modules 目录
  }
]
}
}
rules: [
  {
    test: /\.jsx?$/,
    use: ['babel-loader'],
    exclude: /node_modules/ //排除 node_modules 目录
  }
]
}
}
rules: [
  {
    test: /\.jsx?$/,
    use: ['babel-loader'],
    exclude: /node_modules/ //排除 node_modules 目录
  }
]
}
}
rules: [
  {
    test: /\.jsx?$/,
    use: ['babel-loader'],
    exclude: /node_modules/ //排除 node_modules 目录
  }
]
}
}
rules: [
  {
    test: /\.jsx?$/,
    use: ['babel-loader'],
    exclude: /node_modules/ //排除 node_modules 目录
  }
]
}
}
rules: [
  {
    test: /\.jsx?$/,
    use: ['babel-loader'],
    exclude: /node_modules/ //排除 node_modules 目录
  }
]
}
}
```

```

    }
  ]
}
: [
  {
    test: /\.jsx?$/,
    use: ['babel-loader'],
    exclude: /node_modules/ //排除 node_modules 目录
  }
]
}
}

  {
    test: /\.jsx?$/,
    use: ['babel-loader'],
    exclude: /node_modules/ //排除 node_modules 目录
  }
]
}
}

  {
    test: /\.jsx?$/,
    use: ['babel-loader'],
    exclude: /node_modules/ //排除 node_modules 目录
  }
]
}
}

  test: /\.jsx?$/,
  use: ['babel-loader'],
  exclude: /node_modules/ //排除 node_modules 目录
}
]
}
}

  test: /\.jsx?$/,
  use: ['babel-loader'],
  exclude: /node_modules/ //排除 node_modules 目录
}
]
}
}
test: /\.jsx?$/,
  use: ['babel-loader'],
  exclude: /node_modules/ //排除 node_modules 目录
}
]
}
}
: /\.jsx?$/,
  use: ['babel-loader'],

```

```

        exclude: /node_modules/ //排除 node_modules 目录
      }
    ]
  }
}
/\.jsx?$/,
  use: ['babel-loader'],
  exclude: /node_modules/ //排除 node_modules 目录
}
]
}
}
'
  use: ['babel-loader'],
  exclude: /node_modules/ //排除 node_modules 目录
}
]
}
}
  use: ['babel-loader'],
  exclude: /node_modules/ //排除 node_modules 目录
}
]
}
}
  use: ['babel-loader'],
  exclude: /node_modules/ //排除 node_modules 目录
}
]
}
}
use: ['babel-loader'],
  exclude: /node_modules/ //排除 node_modules 目录
}
]
}
}
: ['babel-loader'],
  exclude: /node_modules/ //排除 node_modules 目录
}
]
}
}
'babel-loader',
  exclude: /node_modules/ //排除 node_modules 目录
}
]
}
}
],
  exclude: /node_modules/ //排除 node_modules 目录
}
]

```



```
}  
}  
}  
}  
}  
}
```

建议给 loader 指定 include 或是 exclude，指定其中一个即可，因为 node\_modules 目录通常不需我们去编译，排除后，有效提升编译效率。

这里，我们可以在 .babelrc 中编写 babel 的配置，也可以在 webpack.config.js 中进行配置。

## 创建一个 .babelrc

配置如下：

```
{  
  "presets": ["@babel/preset-env"],  
  "plugins": [  
    [  
      "@babel/plugin-transform-runtime",  
      {  
        "corejs": 3  
      }  
    ]  
  ]  
}  
  
"presets": ["@babel/preset-env"],  
"plugins": [  
  [  
    "@babel/plugin-transform-runtime",  
    {  
      "corejs": 3  
    }  
  ]  
]  
}  
  
"presets": ["@babel/preset-env"],  
"plugins": [  
  [  
    "@babel/plugin-transform-runtime",  
    {  
      "corejs": 3  
    }  
  ]  
]  
]  
}  
"presets": ["@babel/preset-env"],  
"plugins": [  
  [  
    "@babel/plugin-transform-runtime",  
    {  
      "corejs": 3  
    }  
  ]  
]  
]
```

```

        "@babel/plugin-transform-runtime",
        {
          "corejs": 3
        }
      ]
    ]
  }
  : ["@babel/preset-env"],
  "plugins": [
    [
      "@babel/plugin-transform-runtime",
      {
        "corejs": 3
      }
    ]
  ]
}
"@babel/preset-env"],
"plugins": [
  [
    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ]
]
}
],
"plugins": [
  [
    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ]
]
}
}
"plugins": [
  [
    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ]
]
}
"plugins": [
  [
    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ]
]
}

```



```

]
}
"plugins": [
  [
    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ]
]
}
: [
  [
    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ]
]
}
[
  "@babel/plugin-transform-runtime",
  {
    "corejs": 3
  }
]
}
[
  "@babel/plugin-transform-runtime",
  {
    "corejs": 3
  }
]
}
[
  "@babel/plugin-transform-runtime",
  {
    "corejs": 3
  }
]
}
"@babel/plugin-transform-runtime",
{
  "corejs": 3
}
]
}
"@babel/plugin-transform-runtime",
{
  "corejs": 3
}
]
}
"@babel/plugin-transform-runtime",
{
  "corejs": 3
}
]
}
"@babel/plugin-transform-runtime",
{

```

```
    "corejs": 3
  }
]
},
',
  {
    "corejs": 3
  }
]
}
{
  "corejs": 3
}
]
}
{
  "corejs": 3
}
]
}
"corejs": 3
}
]
}
"corejs": 3
}
]
}
"corejs": 3
}
]
}
: 3
}
]
}
3
}
]
}
]
}
```

```
  ]
}
  ]
  ]
}
  ]
  ]
}
  ]
  ]
}
  ]
  ]
}
}
}
```

现在，我们重新执行 `npx webpack --mode=development`，查看 `dist/main.js`，会发现已经被编译成了低版本的JS代码。

## 在webpack中配置 babel

```
//webpack.config.js
module.exports = {
  // mode: 'development',
  module: {
    rules: [
      {
        test: /\.jsx?$/,
        use: {
          loader: 'babel-loader',
          options: {
            presets: ["@babel/preset-env"],
            plugins: [
              [
                "@babel/plugin-transform-runtime",
                {
                  "corejs": 3
                }
              ]
            ]
          }
        },
        exclude: /node_modules/
      }
    ]
  }
}
```

```

module.exports = {
  // mode: 'development',
  module: {
    rules: [
      {
        test: /\.jsx?$/,
        use: {
          loader: 'babel-loader',
          options: {
            presets: ["@babel/preset-env"],
            plugins: [
              [
                "@babel/plugin-transform-runtime",
                {
                  "corejs": 3
                }
              ]
            ]
          }
        },
        exclude: /node_modules/
      }
    ]
  }
}

module.exports = {
  // mode: 'development',
  module: {
    rules: [
      {
        test: /\.jsx?$/,
        use: {
          loader: 'babel-loader',
          options: {
            presets: ["@babel/preset-env"],
            plugins: [
              [
                "@babel/plugin-transform-runtime",
                {
                  "corejs": 3
                }
              ]
            ]
          }
        },
        exclude: /node_modules/
      }
    ]
  }
}

exports = {
  // mode: 'development',
  module: {

```

```

rules: [
  {
    test: /\.jsx?$/,
    use: {
      loader: 'babel-loader',
      options: {
        presets: ["@babel/preset-env"],
        plugins: [
          [
            "@babel/plugin-transform-runtime",
            {
              "corejs": 3
            }
          ]
        ]
      }
    },
    exclude: /node_modules/
  }
]
}

```

```

// mode: 'development',
module: {
  rules: [
    {
      test: /\.jsx?$/,
      use: {
        loader: 'babel-loader',
        options: {
          presets: ["@babel/preset-env"],
          plugins: [
            [
              "@babel/plugin-transform-runtime",
              {
                "corejs": 3
              }
            ]
          ]
        }
      },
      exclude: /node_modules/
    }
  ]
}
}

```

```

// mode: 'development',
module: {
  rules: [
    {
      test: /\.jsx?$/,
      use: {
        loader: 'babel-loader',

```

```

    options: {
      presets: ["@babel/preset-env"],
      plugins: [
        [
          "@babel/plugin-transform-runtime",
          {
            "corejs": 3
          }
        ]
      ]
    },
    exclude: /node_modules/
  }
]
}
}
// mode: 'development',
module: {
  rules: [
    {
      test: /\.jsx?$/,
      use: {
        loader: 'babel-loader',
        options: {
          presets: ["@babel/preset-env"],
          plugins: [
            [
              "@babel/plugin-transform-runtime",
              {
                "corejs": 3
              }
            ]
          ]
        },
        exclude: /node_modules/
      }
    ]
  }
}

module: {
  rules: [
    {
      test: /\.jsx?$/,
      use: {
        loader: 'babel-loader',
        options: {
          presets: ["@babel/preset-env"],
          plugins: [
            [
              "@babel/plugin-transform-runtime",
              {

```

```

        "corejs": 3
      }
    ]
  ],
},
exclude: /node_modules/
}
]
}
}
module: {
  rules: [
    {
      test: /\.jsx?$/,
      use: {
        loader: 'babel-loader',
        options: {
          presets: ["@babel/preset-env"],
          plugins: [
            [
              "@babel/plugin-transform-runtime",
              {
                "corejs": 3
              }
            ]
          ]
        }
      },
      exclude: /node_modules/
    }
  ]
}
}
module: {
  rules: [
    {
      test: /\.jsx?$/,
      use: {
        loader: 'babel-loader',
        options: {
          presets: ["@babel/preset-env"],
          plugins: [
            [
              "@babel/plugin-transform-runtime",
              {
                "corejs": 3
              }
            ]
          ]
        }
      },
      exclude: /node_modules/
    }
  ]
}
}

```

```

    ]
  }
}
: {
  rules: [
    {
      test: /\.jsx?$/,
      use: {
        loader: 'babel-loader',
        options: {
          presets: ["@babel/preset-env"],
          plugins: [
            [
              "@babel/plugin-transform-runtime",
              {
                "corejs": 3
              }
            ]
          ]
        }
      },
      exclude: /node_modules/
    }
  ]
}
}

rules: [
  {
    test: /\.jsx?$/,
    use: {
      loader: 'babel-loader',
      options: {
        presets: ["@babel/preset-env"],
        plugins: [
          [
            "@babel/plugin-transform-runtime",
            {
              "corejs": 3
            }
          ]
        ]
      },
      exclude: /node_modules/
    }
  ]
}
}

rules: [
  {
    test: /\.jsx?$/,
    use: {
      loader: 'babel-loader',

```



```

    options: {
      presets: ["@babel/preset-env"],
      plugins: [
        [
          "@babel/plugin-transform-runtime",
          {
            "corejs": 3
          }
        ]
      ]
    },
    exclude: /node_modules/
  }
]
}
}
rules: [
  {
    test: /\.jsx?$/,
    use: {
      loader: 'babel-loader',
      options: {
        presets: ["@babel/preset-env"],
        plugins: [
          [
            "@babel/plugin-transform-runtime",
            {
              "corejs": 3
            }
          ]
        ]
      }
    },
    exclude: /node_modules/
  }
]
}
}
: [
  {
    test: /\.jsx?$/,
    use: {
      loader: 'babel-loader',
      options: {
        presets: ["@babel/preset-env"],
        plugins: [
          [
            "@babel/plugin-transform-runtime",
            {
              "corejs": 3
            }
          ]
        ]
      }
    }
  }
]

```

```

    }
  },
  exclude: /node_modules/
}
]
}
}

{
  test: /\.jsx?$/,
  use: {
    loader: 'babel-loader',
    options: {
      presets: ["@babel/preset-env"],
      plugins: [
        [
          "@babel/plugin-transform-runtime",
          {
            "corejs": 3
          }
        ]
      ]
    }
  },
  exclude: /node_modules/
}
]
}
}

{
  test: /\.jsx?$/,
  use: {
    loader: 'babel-loader',
    options: {
      presets: ["@babel/preset-env"],
      plugins: [
        [
          "@babel/plugin-transform-runtime",
          {
            "corejs": 3
          }
        ]
      ]
    }
  },
  exclude: /node_modules/
}
]
}
}

test: /\.jsx?$/,
use: {
  loader: 'babel-loader',

```

```

    options: {
      presets: ["@babel/preset-env"],
      plugins: [
        [
          "@babel/plugin-transform-runtime",
          {
            "corejs": 3
          }
        ]
      ]
    },
    exclude: /node_modules/
  }
]
}
}

test: /\.jsx?$/,
use: {
  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"],
    plugins: [
      [
        "@babel/plugin-transform-runtime",
        {
          "corejs": 3
        }
      ]
    ]
  }
},
exclude: /node_modules/
}
]
}
}

test: /\.jsx?$/,
use: {
  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"],
    plugins: [
      [
        "@babel/plugin-transform-runtime",
        {
          "corejs": 3
        }
      ]
    ]
  }
},
exclude: /node_modules/
}
}
}

```

```

    ]
  }
}
: /\.jsx?$/,
  use: {
    loader: 'babel-loader',
    options: {
      presets: ["@babel/preset-env"],
      plugins: [
        [
          "@babel/plugin-transform-runtime",
          {
            "corejs": 3
          }
        ]
      ]
    },
    exclude: /node_modules/
  }
}
]
}
}
: /\.jsx?$/,
  use: {
    loader: 'babel-loader',
    options: {
      presets: ["@babel/preset-env"],
      plugins: [
        [
          "@babel/plugin-transform-runtime",
          {
            "corejs": 3
          }
        ]
      ]
    },
    exclude: /node_modules/
  }
}
]
}
}
,
  use: {
    loader: 'babel-loader',
    options: {
      presets: ["@babel/preset-env"],
      plugins: [
        [
          "@babel/plugin-transform-runtime",
          {
            "corejs": 3
          }
        ]
      ]
    }
  }
}

```

```

    ]
  ]
}
},
exclude: /node_modules/
}
]
}
}

use: {
  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"],
    plugins: [
      [
        "@babel/plugin-transform-runtime",
        {
          "corejs": 3
        }
      ]
    ]
  },
  exclude: /node_modules/
}
]
}
}

use: {
  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"],
    plugins: [
      [
        "@babel/plugin-transform-runtime",
        {
          "corejs": 3
        }
      ]
    ]
  },
  exclude: /node_modules/
}
]
}
}

use: {
  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"],
    plugins: [
      [

```

```

        "@babel/plugin-transform-runtime",
        {
          "corejs": 3
        }
      ]
    }
  },
  exclude: /node_modules/
}
]
}
}
: {
  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"],
    plugins: [
      [
        "@babel/plugin-transform-runtime",
        {
          "corejs": 3
        }
      ]
    ]
  },
  exclude: /node_modules/
}
]
}
}

  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"],
    plugins: [
      [
        "@babel/plugin-transform-runtime",
        {
          "corejs": 3
        }
      ]
    ]
  },
  exclude: /node_modules/
}
]
}
}

  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"],

```

```

    plugins: [
      [
        "@babel/plugin-transform-runtime",
        {
          "corejs": 3
        }
      ]
    ]
  },
  exclude: /node_modules/
}
]
}
}
loader: 'babel-loader',
options: {
  presets: ["@babel/preset-env"],
  plugins: [
    [
      "@babel/plugin-transform-runtime",
      {
        "corejs": 3
      }
    ]
  ]
},
exclude: /node_modules/
}
]
}
}
: 'babel-loader',
options: {
  presets: ["@babel/preset-env"],
  plugins: [
    [
      "@babel/plugin-transform-runtime",
      {
        "corejs": 3
      }
    ]
  ]
},
exclude: /node_modules/
}
]
}
}
'babel-loader',
options: {
  presets: ["@babel/preset-env"],

```

```

    plugins: [
      [
        "@babel/plugin-transform-runtime",
        {
          "corejs": 3
        }
      ]
    ]
  },
  exclude: /node_modules/
}
]
}
}
,
options: {
  presets: ["@babel/preset-env"],
  plugins: [
    [
      "@babel/plugin-transform-runtime",
      {
        "corejs": 3
      }
    ]
  ]
},
exclude: /node_modules/
}
]
}
}
}
options: {
  presets: ["@babel/preset-env"],
  plugins: [
    [
      "@babel/plugin-transform-runtime",
      {
        "corejs": 3
      }
    ]
  ]
},
exclude: /node_modules/
}
]
}
}
}
options: {
  presets: ["@babel/preset-env"],
  plugins: [

```



```

    [
      "@babel/plugin-transform-runtime",
      {
        "corejs": 3
      }
    ]
  ],
},
exclude: /node_modules/
}
]
}
}
options: {
  presets: ["@babel/preset-env"],
  plugins: [
    [
      "@babel/plugin-transform-runtime",
      {
        "corejs": 3
      }
    ]
  ],
  exclude: /node_modules/
}
]
}
}
: {
  presets: ["@babel/preset-env"],
  plugins: [
    [
      "@babel/plugin-transform-runtime",
      {
        "corejs": 3
      }
    ]
  ],
  exclude: /node_modules/
}
]
}
}
}
  presets: ["@babel/preset-env"],
  plugins: [
    [
      "@babel/plugin-transform-runtime",
      {

```

```

    "corejs": 3
  }
]
},
exclude: /node_modules/
}
]
}
}
}

presets: ["@babel/preset-env"],
plugins: [
  [
    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ]
],
exclude: /node_modules/
}
]
}
}
presets: ["@babel/preset-env"],
plugins: [
  [
    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ]
],
exclude: /node_modules/
}
]
}
}
}
presets: ["@babel/preset-env"],
plugins: [
  [
    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ]
],
}
],
}
},

```

```

        exclude: /node_modules/
      }
    ]
  }
}
"@babel/preset-env",
  plugins: [
    [
      "@babel/plugin-transform-runtime",
      {
        "corejs": 3
      }
    ]
  ]
},
exclude: /node_modules/
}
]
}
},
],
  plugins: [
    [
      "@babel/plugin-transform-runtime",
      {
        "corejs": 3
      }
    ]
  ]
},
exclude: /node_modules/
}
]
}
}
  plugins: [
    [
      "@babel/plugin-transform-runtime",
      {
        "corejs": 3
      }
    ]
  ]
},
exclude: /node_modules/
}
]
}
}
  plugins: [

```

```

        [
          "@babel/plugin-transform-runtime",
          {
            "corejs": 3
          }
        ]
      ],
    },
    exclude: /node_modules/
  }
]
}
}
plugins: [
  [
    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ]
],
exclude: /node_modules/
}
]
}
}
: [
  [
    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ]
],
exclude: /node_modules/
}
]
}
}

  [
    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ]
],

```

```

    exclude: /node_modules/
  }
]
}
}

    [
      "@babel/plugin-transform-runtime",
      {
        "corejs": 3
      }
    ]
  },
  exclude: /node_modules/
}
]
}
}

    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ],
  exclude: /node_modules/
}
]
}
}

    "@babel/plugin-transform-runtime",
    {
      "corejs": 3
    }
  ],
  exclude: /node_modules/
}
]
}
}

"@babel/plugin-transform-runtime",
  {
    "corejs": 3
  }
]
},

```

```
    exclude: /node_modules/
  }
]
},
',
    {
      "corejs": 3
    }
  ]
}
},
exclude: /node_modules/
}
]
}
}
},
    {
      "corejs": 3
    }
  ]
}
},
exclude: /node_modules/
}
]
}
}
},
    {
      "corejs": 3
    }
  ]
}
},
exclude: /node_modules/
}
]
}
}
},
    "corejs": 3
  }
]
}
},
exclude: /node_modules/
}
]
}
```

```

}
      "corejs": 3
    }
  ],
},
exclude: /node_modules/
}
]
}
}
}
"corejs": 3
  "corejs": 3
    "corejs": 3
  ],
},
exclude: /node_modules/
}
]
}
}
}
: 3
  "corejs": 3
    "corejs": 3
  ],
},
exclude: /node_modules/
}
]
}
}
}
3
  "corejs": 3
    "corejs": 3
  ],
},
exclude: /node_modules/
}
]
}
}
}
}
}
  "corejs": 3
    "corejs": 3
  ],
},
exclude: /node_modules/
}
]
}
}
}
}
]

```

```
}
}
}
}
},
exclude: /node_modules/
}
}
}
},
exclude: /node_modules/
}
}
}
},
exclude: /node_modules/
}
}
}
},
exclude: /node_modules/
}
}
},
exclude: /node_modules/
}
}
},
exclude: /node_modules/
}
```



```
}
}
    }
    },
    exclude: /node_modules/
  }
]
}
}
    },
    exclude: /node_modules/
  }
]
}
}
    },
    exclude: /node_modules/
  }
]
}
}
    exclude: /node_modules/
  }
]
}
}
    exclude: /node_modules/
  }
]
}
}
exclude: /node_modules/
}
]
}
}
: /node_modules/
}
]
}
}
/node_modules/
}
]
}
}
}
}
}
}
}
}
}
```

```
    ]
  }
}

    ]
  }
}

    ]
  }
}

}
}
}

}
}
```

这里有几点需要说明：

- loader 需要配置在 `module.rules` 中，`rules` 是一个数组。
- loader 的格式为：

```
{
  test: /\.jsx?$/,//匹配规则
  use: 'babel-loader'
}

  test: /\.jsx?$/,//匹配规则
  use: 'babel-loader'
}
  test: /\.jsx?$/,//匹配规则
  use: 'babel-loader'
}
test: /\.jsx?$/,//匹配规则
  use: 'babel-loader'
}
: /\.jsx?$/,//匹配规则
  use: 'babel-loader'
}
\./jsx?$/,//匹配规则
  use: 'babel-loader'
}
,//匹配规则
  use: 'babel-loader'
}
//匹配规则
  use: 'babel-loader'
}

  use: 'babel-loader'
}
```

```
    use: 'babel-loader'
  }
  'babel-loader'
}
}
}
```

或者也可以像下面这样:

//适用于只有一个 loader 的情况

```
{
  test: /\.jsx?$/,
  loader: 'babel-loader',
  options: {
    //...
  }
}

{
  test: /\.jsx?$/,
  loader: 'babel-loader',
  options: {
    //...
  }
}

{
  test: /\.jsx?$/,
  loader: 'babel-loader',
  options: {
    //...
  }
}

  test: /\.jsx?$/,
  loader: 'babel-loader',
  options: {
    //...
  }
}

  test: /\.jsx?$/,
  loader: 'babel-loader',
  options: {
    //...
  }
}

test: /\.jsx?$/,
  loader: 'babel-loader',
  options: {
    //...
  }
}
: /\.jsx?$/,
```

```

    loader: 'babel-loader',
    options: {
      //...
    }
  }
  /\.jsx?$/,
  loader: 'babel-loader',
  options: {
    //...
  }
},
  loader: 'babel-loader',
  options: {
    //...
  }
}

  loader: 'babel-loader',
  options: {
    //...
  }
}
  loader: 'babel-loader',
  options: {
    //...
  }
}
loader: 'babel-loader',
  options: {
    //...
  }
}
: 'babel-loader',
  options: {
    //...
  }
}
'babel-loader',
  options: {
    //...
  }
}
,
  options: {
    //...
  }
}

  options: {
    //...
  }
}
}
  options: {

```



```

{
  test: /\.jsx?$/,
  use: {
    loader: 'babel-loader',
    options: {
      presets: ["@babel/preset-env"]
    }
  },
  exclude: /node_modules/
}
]
{
  test: /\.jsx?$/,
  use: {
    loader: 'babel-loader',
    options: {
      presets: ["@babel/preset-env"]
    }
  },
  exclude: /node_modules/
}
]

test: /\.jsx?$/,
use: {
  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"]
  }
},
exclude: /node_modules/
}
]

test: /\.jsx?$/,
use: {
  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"]
  }
},
exclude: /node_modules/
}
]

test: /\.jsx?$/,
use: {
  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"]
  }
},
exclude: /node_modules/
}
]
: /\.jsx?$/,

```

```

    use: {
      loader: 'babel-loader',
      options: {
        presets: ["@babel/preset-env"]
      }
    },
    exclude: /node_modules/
  }
]
 /\.jsx?$/,
  use: {
    loader: 'babel-loader',
    options: {
      presets: ["@babel/preset-env"]
    }
  },
  exclude: /node_modules/
}
]
',
  use: {
    loader: 'babel-loader',
    options: {
      presets: ["@babel/preset-env"]
    }
  },
  exclude: /node_modules/
}
]

  use: {
    loader: 'babel-loader',
    options: {
      presets: ["@babel/preset-env"]
    }
  },
  exclude: /node_modules/
}
]

  use: {
    loader: 'babel-loader',
    options: {
      presets: ["@babel/preset-env"]
    }
  },
  exclude: /node_modules/
}
]

  use: {
    loader: 'babel-loader',
    options: {
      presets: ["@babel/preset-env"]
    }
  },
  exclude: /node_modules/
}
]
]
use: {
  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"]
  }
},

```

```

    exclude: /node_modules/
  }
]
: {
  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"]
  }
},
exclude: /node_modules/
}
]

  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"]
  }
},
exclude: /node_modules/
}
]

  loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"]
  }
},
exclude: /node_modules/
}
]
loader: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"]
  }
},
exclude: /node_modules/
}
]
: 'babel-loader',
  options: {
    presets: ["@babel/preset-env"]
  }
},
exclude: /node_modules/
}
]
'babel-loader',
  options: {
    presets: ["@babel/preset-env"]
  }
},
exclude: /node_modules/
}
]
,

```



```

    options: {
      presets: ["@babel/preset-env"]
    }
  },
  exclude: /node_modules/
}
]
    options: {
      presets: ["@babel/preset-env"]
    }
  },
  exclude: /node_modules/
}
]
    options: {
      presets: ["@babel/preset-env"]
    }
  },
  exclude: /node_modules/
}
]
options: {
  presets: ["@babel/preset-env"]
}
},
exclude: /node_modules/
}
]
: {
  presets: ["@babel/preset-env"]
}
},
exclude: /node_modules/
}
]
    presets: ["@babel/preset-env"]
  },
  exclude: /node_modules/
}
]
    presets: ["@babel/preset-env"]
  },
  exclude: /node_modules/
}
]
presets: ["@babel/preset-env"]
}
},
exclude: /node_modules/
}

```



```
]
}
]
}
]
]
]
]
```

上面我们说了如何将JS的代码编译成向下兼容的代码，当然你还可以还需要一些其它的 `babel` 的插件和设，例如 `@babel/preset-react`，`@babel/plugin-proposal-optional-chaining` 等，不过，`babel` 配置并非本文的重点，我们继续往下。

不要说细心的小伙伴了，即使是粗心的小伙伴肯定也发现了，我们在使用 `webpack` 进行打包的时候一直运行的都是 `npx webpack --mode=development` 是否可以将 `mode` 配置在 `webpack.config.js` 中呢？显然是可以的。

## 5.mode

将 `mode` 增加到 `webpack.config.js` 中:

```
module.exports = {
  //....
  mode: "development",
  module: {
    //...
  }
}
.exports = {
  //....
  mode: "development",
  module: {
    //...
  }
}

//....
mode: "development",
module: {
  //...
}
}

//....
mode: "development",
module: {
  //...
}
}

//....
mode: "development",
module: {
  //...
```

```
    }
  }

  mode: "development",
  module: {
    //...
  }
}
mode: "development",
module: {
  //...
}
}
"development",
module: {
  //...
}
}
'
  module: {
    //...
  }
}

  module: {
    //...
  }
}
  module: {
    //...
  }
}
module: {
  //...
}
}
: {
  //...
}
}

  //...
}
}

  //...
}
}
//...
}
}

}
}
}
```

```
}  
  
}  
}
```

`mode` 配置项, 告知 `webpack` 使用相应模式的内置优化。

`mode` 配置项, 支持以下两个配置:

- `development`: 将 `process.env.NODE_ENV` 的值设置为 `development`, 启用 `NamedChunksPlugin` 和 `NamedModulesPlugin`
- `production`: 将 `process.env.NODE_ENV` 的值设置为 `production`, 启用 `FlagDependencyUsagePlugin`, `FlagIncludedChunksPlugin`, `ModuleConcatenationPlugin`, `NoEmitOnErrorsPlugin`, `OccurrenceOrderPlugin`, `SideEffectsFlagPlugin` 和 `UglifyJsPlugin`

现在, 我们之间使用 `npx webpack` 进行编译即可。

## 6.在浏览器中查看页面

搞了这么久, 还不能在浏览器中查看页面, 这显然不能忍!

查看页面, 难免就需要 `html` 文件, 有小伙伴可能知道, 有时我们会指定打包文件中带有 `hash`, 那么次生成的 `js` 文件名会有所不同, 总不能让我们每次都人工去修改 `html`, 这样不是显得我们很蠢嘛~

我们可以使用 `html-webpack-plugin` 插件来帮助我们完成这些事情。

首先, 安装一下插件:

```
npm install html-webpack-plugin -D
```

新建 `public` 目录, 并在其中新建一个 `index.html` 文件( 文件内容使用 `html5` 快捷生成即可)

修改 `webpack.config.js` 文件。

```
//首先引入插件  
const HtmlWebpackPlugin = require('html-webpack-plugin');  
module.exports = {  
  //...  
  plugins: [  
    //数组 放着所有的webpack插件  
    new HtmlWebpackPlugin({  
      template: './public/index.html',  
      filename: 'index.html', //打包后的文件名  
      minify: {  
        removeAttributeQuotes: false, //是否删除属性的双引号  
        collapseWhitespace: false, //是否折叠空白  
      },  
      // hash: true //是否加上hash, 默认是 false  
    })  
  ]  
}
```

```

const HtmlWebpackPlugin = require('html-webpack-plugin');
module.exports = {
  //...
  plugins: [
    //数组 放着所有的webpack插件
    new HtmlWebpackPlugin({
      template: './public/index.html',
      filename: 'index.html', //打包后的文件名
      minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
      },
      // hash: true //是否加上hash, 默认是 false
    })
  ]
}

const HtmlWebpackPlugin = require('html-webpack-plugin');
module.exports = {
  //...
  plugins: [
    //数组 放着所有的webpack插件
    new HtmlWebpackPlugin({
      template: './public/index.html',
      filename: 'index.html', //打包后的文件名
      minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
      },
      // hash: true //是否加上hash, 默认是 false
    })
  ]
}

HtmlWebpackPlugin = require('html-webpack-plugin');
module.exports = {
  //...
  plugins: [
    //数组 放着所有的webpack插件
    new HtmlWebpackPlugin({
      template: './public/index.html',
      filename: 'index.html', //打包后的文件名
      minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
      },
      // hash: true //是否加上hash, 默认是 false
    })
  ]
}

require('html-webpack-plugin');
module.exports = {
  //...
  plugins: [
    //数组 放着所有的webpack插件
    new HtmlWebpackPlugin({

```

```

    template: './public/index.html',
    filename: 'index.html', //打包后的文件名
    minify: {
      removeAttributeQuotes: false, //是否删除属性的双引号
      collapseWhitespace: false, //是否折叠空白
    },
    // hash: true //是否加上hash, 默认是 false
  })
]
}
('html-webpack-plugin');
module.exports = {
  //...
  plugins: [
    //数组 放着所有的webpack插件
    new HtmlWebpackPlugin({
      template: './public/index.html',
      filename: 'index.html', //打包后的文件名
      minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
      },
      // hash: true //是否加上hash, 默认是 false
    })
  ]
}
'html-webpack-plugin');
module.exports = {
  //...
  plugins: [
    //数组 放着所有的webpack插件
    new HtmlWebpackPlugin({
      template: './public/index.html',
      filename: 'index.html', //打包后的文件名
      minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
      },
      // hash: true //是否加上hash, 默认是 false
    })
  ]
}
);
module.exports = {
  //...
  plugins: [
    //数组 放着所有的webpack插件
    new HtmlWebpackPlugin({
      template: './public/index.html',
      filename: 'index.html', //打包后的文件名
      minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
      },
    },
  ],
}

```

```

        // hash: true //是否加上hash, 默认是 false
    })
  ]
}

module.exports = {
  //...
  plugins: [
    //数组 放着所有的webpack插件
    new HtmlWebpackPlugin({
      template: './public/index.html',
      filename: 'index.html', //打包后的文件名
      minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
      },
      // hash: true //是否加上hash, 默认是 false
    })
  ]
}

module.exports = {
  //...
  plugins: [
    //数组 放着所有的webpack插件
    new HtmlWebpackPlugin({
      template: './public/index.html',
      filename: 'index.html', //打包后的文件名
      minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
      },
      // hash: true //是否加上hash, 默认是 false
    })
  ]
}

.exports = {
  //...
  plugins: [
    //数组 放着所有的webpack插件
    new HtmlWebpackPlugin({
      template: './public/index.html',
      filename: 'index.html', //打包后的文件名
      minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
      },
      // hash: true //是否加上hash, 默认是 false
    })
  ]
}

//...
plugins: [
  //数组 放着所有的webpack插件

```



```

new HtmlWebpackPlugin({
  template: './public/index.html',
  filename: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
//...
plugins: [
  //数组 放着所有的webpack插件
  new HtmlWebpackPlugin({
    template: './public/index.html',
    filename: 'index.html', //打包后的文件名
    minify: {
      removeAttributeQuotes: false, //是否删除属性的双引号
      collapseWhitespace: false, //是否折叠空白
    },
    // hash: true //是否加上hash, 默认是 false
  })
]
}
//...
plugins: [
  //数组 放着所有的webpack插件
  new HtmlWebpackPlugin({
    template: './public/index.html',
    filename: 'index.html', //打包后的文件名
    minify: {
      removeAttributeQuotes: false, //是否删除属性的双引号
      collapseWhitespace: false, //是否折叠空白
    },
    // hash: true //是否加上hash, 默认是 false
  })
]
}

plugins: [
  //数组 放着所有的webpack插件
  new HtmlWebpackPlugin({
    template: './public/index.html',
    filename: 'index.html', //打包后的文件名
    minify: {
      removeAttributeQuotes: false, //是否删除属性的双引号
      collapseWhitespace: false, //是否折叠空白
    },
    // hash: true //是否加上hash, 默认是 false
  })
]
}
plugins: [

```

```
//数组 放着所有的webpack插件
new HtmlWebpackPlugin({
  template: './public/index.html',
  filename: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
```

```
//数组 放着所有的webpack插件
new HtmlWebpackPlugin({
  template: './public/index.html',
  filename: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
```

```
//数组 放着所有的webpack插件
new HtmlWebpackPlugin({
  template: './public/index.html',
  filename: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
```

```
//数组 放着所有的webpack插件
new HtmlWebpackPlugin({
  template: './public/index.html',
  filename: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
```

```
new HtmlWebpackPlugin({
  template: './public/index.html',
  filename: 'index.html', //打包后的文件名
  minify: {
```

```

        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
    },
    // hash: true //是否加上hash, 默认是 false
})
]
}
new HtmlWebpackPlugin({
    template: './public/index.html',
    filename: 'index.html', //打包后的文件名
    minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
    },
    // hash: true //是否加上hash, 默认是 false
})
]
}
new HtmlWebpackPlugin({
    template: './public/index.html',
    filename: 'index.html', //打包后的文件名
    minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
    },
    // hash: true //是否加上hash, 默认是 false
})
]
}
HtmlWebpackPlugin({
    template: './public/index.html',
    filename: 'index.html', //打包后的文件名
    minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
    },
    // hash: true //是否加上hash, 默认是 false
})
]
}

    template: './public/index.html',
    filename: 'index.html', //打包后的文件名
    minify: {
        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
    },
    // hash: true //是否加上hash, 默认是 false
})
]
}

    template: './public/index.html',
    filename: 'index.html', //打包后的文件名
    minify: {

```

```

        removeAttributeQuotes: false, //是否删除属性的双引号
        collapseWhitespace: false, //是否折叠空白
    },
    // hash: true //是否加上hash, 默认是 false
})
]
}
template: './public/index.html',
  filename: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
: './public/index.html',
  filename: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
'./public/index.html',
  filename: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
'
  filename: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}

  filename: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false

```

```

    })
  ]
}
  filename: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
filename: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
: 'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
'index.html', //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
, //打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
//打包后的文件名
  minify: {
    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false

```

```

    })
  ]
}

    minify: {
      removeAttributeQuotes: false, //是否删除属性的双引号
      collapseWhitespace: false, //是否折叠空白
    },
    // hash: true //是否加上hash, 默认是 false
  })
]
}

    minify: {
      removeAttributeQuotes: false, //是否删除属性的双引号
      collapseWhitespace: false, //是否折叠空白
    },
    // hash: true //是否加上hash, 默认是 false
  })
]
}

    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}

    removeAttributeQuotes: false, //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}

removeAttributeQuotes: false, //是否删除属性的双引号
collapseWhitespace: false, //是否折叠空白
},
// hash: true //是否加上hash, 默认是 false
})
]
}

: false, //是否删除属性的双引号
collapseWhitespace: false, //是否折叠空白
},
// hash: true //是否加上hash, 默认是 false
})
]
}

false, //是否删除属性的双引号
collapseWhitespace: false, //是否折叠空白
},
// hash: true //是否加上hash, 默认是 false
})

```

```

    ]
  }
  //是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
//是否删除属性的双引号
    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}

    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}

    collapseWhitespace: false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
false, //是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}
//是否折叠空白
  },
  // hash: true //是否加上hash, 默认是 false
})
]
}

    },
  // hash: true //是否加上hash, 默认是 false
})
]
}
  },

```

```
        // hash: true //是否加上hash, 默认是 false
    })
  ]
}

        // hash: true //是否加上hash, 默认是 false
    })
  ]
}

        // hash: true //是否加上hash, 默认是 false
    })
  ]
}
// hash: true //是否加上hash, 默认是 false
})
]
}

})
]
}

})
]
}

]
}
]
}

}
}
```

此时执行 `npm run webpack`, 可以看到 `dist` 目录下新增了 `index.html` 文件, 并且其中自动插入了 `<script>` 脚本, 引入的是我们打包之后的 `js` 文件。

这里要多说一点点东西, `HtmlWebpackPlugin` 还为我们提供了一个 `config` 的配置, 这个配置可以是非常有用了。

由于帖子 102400 长度限制被截断了。

语雀: <https://www.yuque.com/fest/articles/psv4gi>

公众号: 「前端时空」

参考资料

- Peer Dependencies
- html-webpack-plugin
- webpack中文文档