

02

OPEN ORIENTED

凹凸实验室

# Webpack打包原理

Simba



为什么要代码分模块、分文件？

为什么要打包？

打包工具都要做些什么？

打包模块

处理依赖

其它功能

# 打包模块

# webpack是如何打包模块的？

入口文件：index.js →



```
index.js
1 console.log('index')
```

Line 1, Column 21      Tab Size: 2      JavaScript

# webpack是如何打包模块的？

webpack build



```
index.js
1  /*****/ (function(modules) { // webpackBootstrap
2  /*****/ // The module cache
3  /*****/ var installedModules = {};
4  /*****/
5  /*****/ // The require function
6  /*****/ function __webpack_require__(moduleId) {
7  /*****/
8  /*****/ // Check if module is in cache
9  /*****/ if(installedModules[moduleId]) {
10 /*****/ return installedModules[moduleId].exports;
11 /*****/ }
12 /*****/ // Create a new module (and put it into the cache)
13 /*****/ var module = installedModules[moduleId] = {
14 /*****/   i: moduleId,
15 /*****/   l: false,
16 /*****/   exports: {}
17 /*****/ };
18 /*****/
19 /*****/ // Execute the module function
20 /*****/ modules[moduleId].call(module.exports, module, module.exp
21 /*****/
22 /*****/ // Flag the module as loaded
23 /*****/ module.l = true;
24 /*****/
25 /*****/ // Return the exports of the module
26 /*****/ return module.exports;
27 /*****/ }
28 /*****/
29 /*****/
30 /*****/ // expose the modules object (__webpack_modules__)
31 /*****/ __webpack_require__.m = modules;
32 /*****/
33 /*****/ // expose the module cache
34 /*****/ __webpack_require__.c = installedModules;
35 /*****/
36 /*****/ // define getter function for harmony exports
37 /*****/ __webpack_require__.d = function(exports, name, getter) {
38 /*****/   if(!__webpack_require__.o(exports, name)) {
39 /*****/     Object.defineProperty(exports, name, {
40 /*****/       configurable: false,
41 /*****/       enumerable: true,
42 /*****/       get: getter
43 /*****/     });
44 /*****/   }
45 /*****/ }
```

# webpack是如何打包模块的?

```
index.js
1  /***/ (function(modules) { // webpackBootstrap
2  /***/   // The module cache
3  /***/   var installedModules = {};
4  /***/   // The require function
5  /***/   function __webpack_require__(moduleId) {
6  /***/     // Check if module is in cache
7  /***/     if(installedModules[moduleId]) {
8  /***/       return installedModules[moduleId].exports;
9  /***/     }
10  /***/     // Create a new module (and put it into the cache)
11  /***/     var module = installedModules[moduleId] = {
12  /***/       i: moduleId,
13  /***/       l: false,
14  /***/       exports: {}
15  /***/     };
16  /***/     // Execute the module function
17  /***/     modules[moduleId].call(module.exports, module, module
18  /***/     // Return the exports of the module
19  /***/     return module.exports;
20  /***/   }
21  /***/   // Load entry module and return exports
22  /***/   return __webpack_require__(__webpack_require__.s = 0);
23  /***/ })
24  /***/
25  /***/ ([
26  /***/   /* 0 */
27  /***/   (function(module, exports) {
28  /***/
29  /***/     console.log('index')
30  /***/
31  /***/   })
32  /***/ ]);
```

→ module

ASCII, Line 28, Column 1; Copied 1 character      Tab Size: 2      JavaScript

webpackBootstrap()  
↓  
\_\_webpack\_require\_\_()  
↓  
module()  
↓  
return module.exports;

# 处理依赖



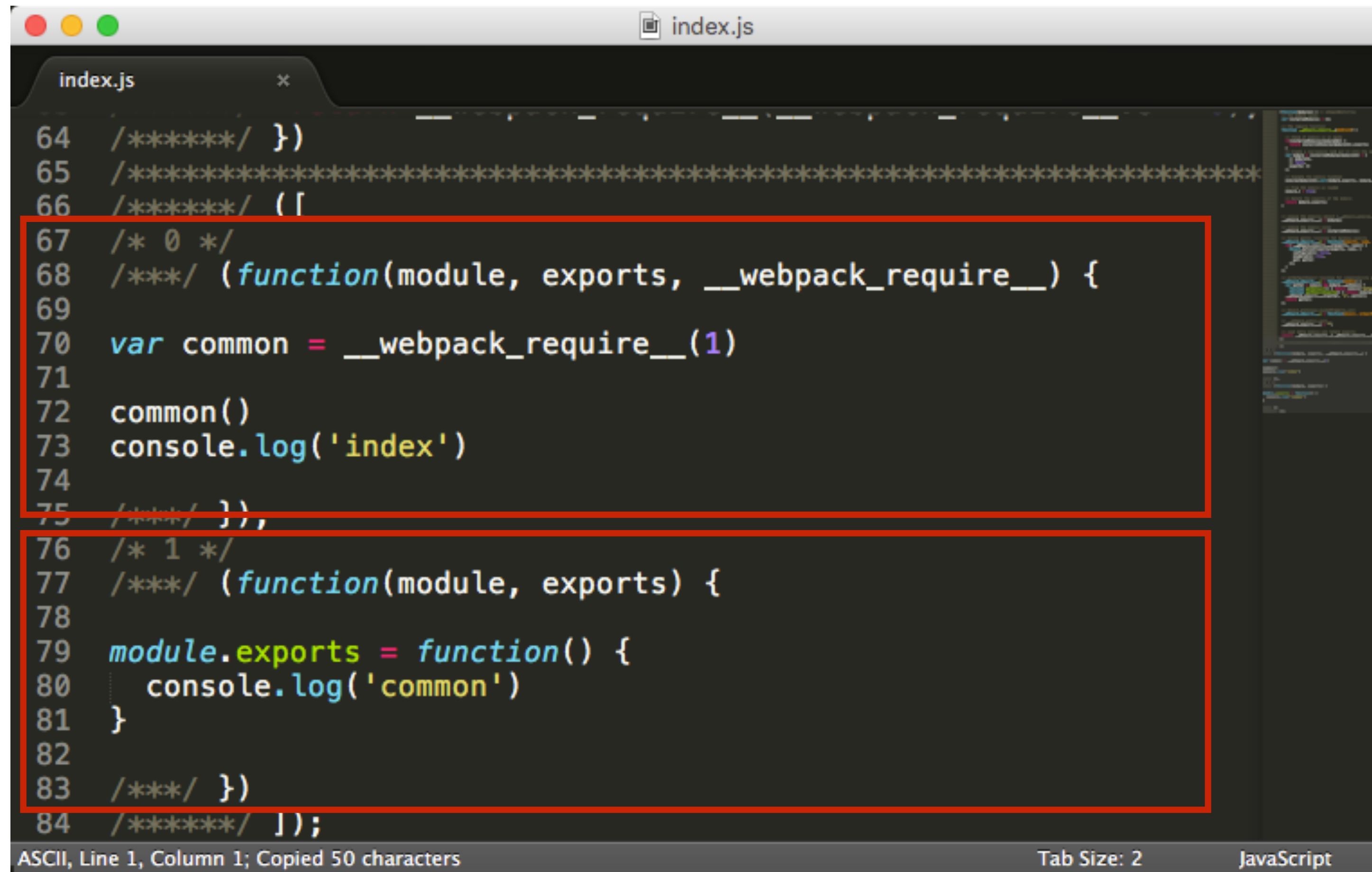
# webpack是如何处理依赖的？

入口文件：index.js →

A screenshot of a code editor window titled 'index.js'. The editor shows four lines of JavaScript code: 1. 'var common = require('../common/common.js')', 2. an empty line, 3. 'common()', and 4. 'console.log('index')'. The fourth line is highlighted with a light blue background. The editor has a dark theme and a tab labeled 'index.js' at the top. The status bar at the bottom indicates 'ASCII, Line 4, Column 21', 'Tab Size: 2', and 'JavaScript'.

```
1  var common = require('../common/common.js')
2
3  common()
4  console.log('index')
```

# webpack是如何处理依赖的?



```
64  /*****/ } )
65  /*****/
66  /*****/ ( [
67  /* 0 */
68  /***/ (function(module, exports, __webpack_require__) {
69
70  var common = __webpack_require__(1)
71
72  common()
73  console.log('index')
74
75  /***/ }),
76  /* 1 */
77  /***/ (function(module, exports) {
78
79  module.exports = function() {
80    console.log('common')
81  }
82
83  /***/ })
84  /*****/ ] );
```

→ modules[0]

→ modules[1]

ASCII, Line 1, Column 1; Copied 50 characters      Tab Size: 2      JavaScript

使用loader



# 依赖一个.jpg图片

入口文件: index.js

A screenshot of a code editor window titled 'index.js'. The code is as follows:

```
1 require('../css/index.css')
2 var sz = require('../images/sz.jpg')
3 var common = require('../common/common.js')
4
5 common()
6 console.log('index')
7 console.log(sz)
```

The editor has a dark theme. The status bar at the bottom shows 'ASCII, Line 4, Column 1', 'Tab Size: 2', and 'JavaScript'.

# 依赖一个.jpg图片

url-loader



```
/* 6 */  
/***/ (function(module, exports, __webpack_require__) {  
module.exports = __webpack_require__.p + "/images/sz.jpg?18lftoX";  
/***/ }),  
/* 7 */
```



图片资源会被build到对应的目录

# 依赖一个.css文件

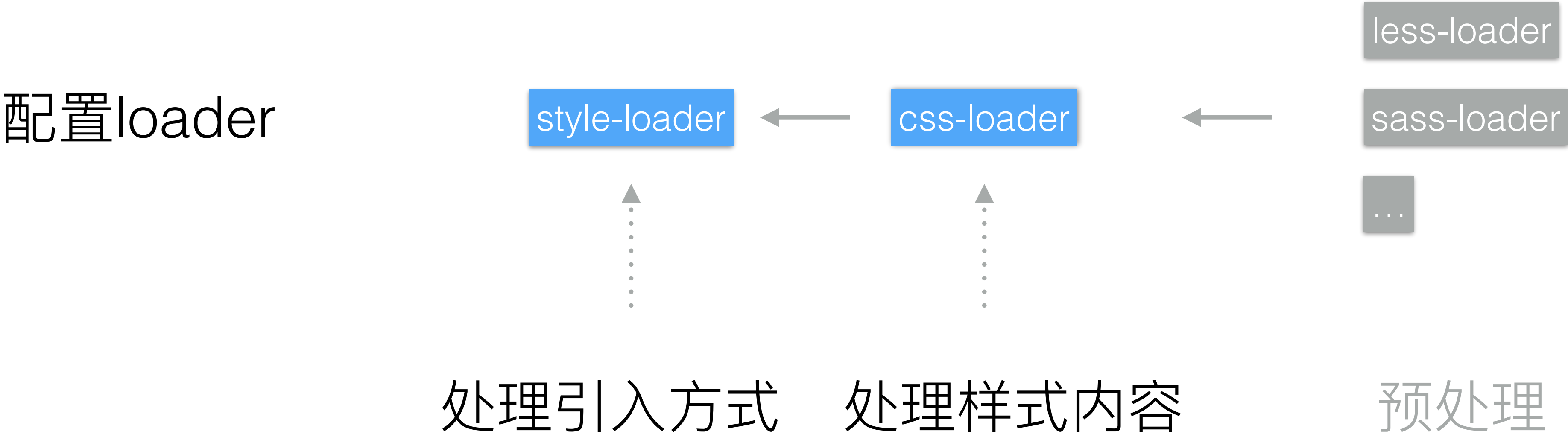
入口文件：index.js →

A screenshot of a code editor window titled 'index.js'. The editor shows five lines of JavaScript code. Line 1: require('../css/index.css'). Line 2: var common = require('../common/common.js'). Line 3: (empty). Line 4: common(). Line 5: console.log('index'). The status bar at the bottom indicates 'ASCII, Line 1, Column 28', 'Tab Size: 2', and 'JavaScript'.

```
1 require('../css/index.css')
2 var common = require('../common/common.js')
3
4 common()
5 console.log('index')
```



# 通过loader来处理依赖



# 通过loader来处理依赖

```
Hash: d908fac174ff2d1f409e
Version: webpack 3.3.0
Time: 556ms

   Asset      Size  Chunks             Chunk Names
index.js  18.3 kB      0  [emitted]  index
   [0] ./src/js/index.js 102 bytes {0} [built]
   [1] ./src/css/index.css 1.01 kB {0} [built]
   [2] ./node_modules/css-loader!./src/css/index.css 188 bytes {0} [built]
   [6] ./src/common/common.js 96 bytes {0} [built]
   [7] ./src/lib/lib.js 51 bytes {0} [built]
+ 3 hidden modules

tapable._plugins = {
  "something": [
    function(a, cb){
      setTimeout(()=>{
        console.log('1', a
        cb();
      }, 1500);
    },
  ],
}
```

用于处理样式内容的三个隐藏的模块将会被打包到index.js中

# 使用插件



# 通过插件处理打包的内容

UglifyJs

`webpack.optimize.UglifyJsPlugin`

CommonsChunk

`webpack.optimize.CommonsChunkPlugin`

ExtractCSS

`extract-text-webpack-plugin`

▪  
▪  
▪

**T H A N K S**  
**FOR YOUR WATCHING**

