Node.js

模块和包

module - 模块

- JavaScript 模块化的意义
    - 大型工程，团队开发：更好的代码复用和维护
    - 解决命名冲突、文件依赖、加载顺序等问题
    - 对比前端模块： RequireJS 、 SeaJS
    - CommonJS 规范

module - 模块

- 思考：编写一个数学运算库

module - 模块

- 前端JS代码如何避免局部变量污染全局?

```
(function(r) {
  var pi = Math.PI;
  return pi * r *r;
})(5);

console.log('PI = ' + pi);
```

module - 模块

- 模块分类
  - 原生的核心模块：fs、http、url …
  - 文件模块
- 如何模块化
  - 一个文件对应一个模块
  - 关键字：require exports module

module - 模块

```
// 模块定义文件 circle.js
var PI = Math.PI; // 私有/局部
exports.area = function(r) {
  return PI * r * r;
};
exports.circumference = function(r) {
  return 2 * PI * r;
};
```

```
// 使用文件 app.js
const circle = require('./circle.js');
console.log('半径为4的圆面积是: ' + circle.area(4));
```

module - 模块

```
// 模块定义文件 circle.js
var PI = Math.PI; // 私有/局部

module.exports = function(r) {
  return {
    area: function() {
      return PI * r * r;
    },
    circumference: function() {
      return 2 * PI * r;
    }
  };
}
```

```
// 使用文件 app.js
const circle = require('./circle.js');
var myCircle = circle(2);
console.log('半径为4的圆面积是: ' + myCircle.area());
```

module - 模块

- 关键字
  - require、module、exports
  - require 是 Node 全局关键字
  - module 和 exports 是模块里的关键字
  - module 对象有个 exports 成员： module.exports = { }
  - exports 只是引用，它指向 module.exports: exports = module.exports

module - 模块

- module 是一个对象，它的一些属性：
  - exports
  - id
  - filename
  - …

```
(function (exports, require, module, __filename, __dirname) {

  module.exports = {};
  exports = module.exports;

  // 你的模块代码放在这里

});
```

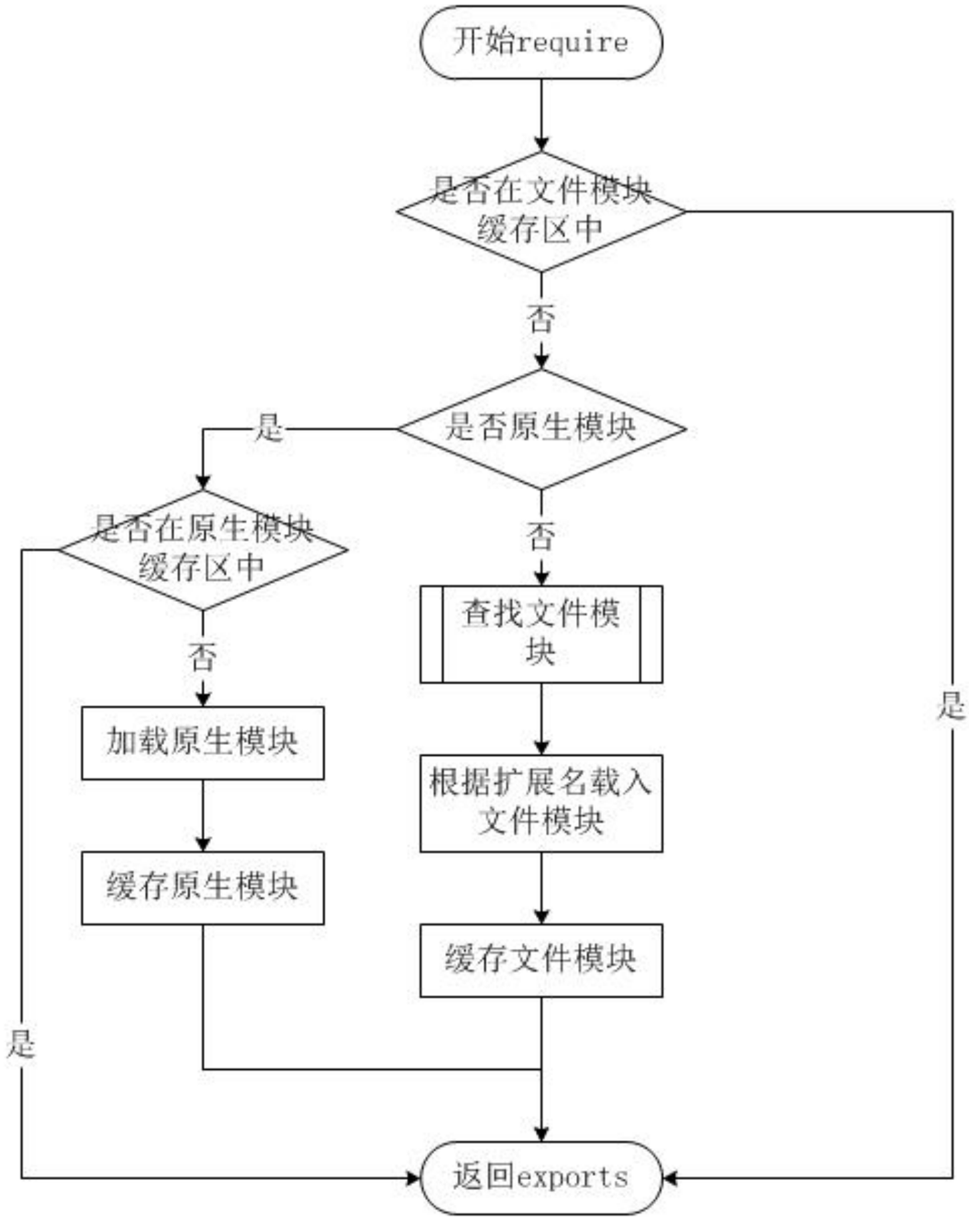
```
(function(r) {
  return Math.PI * r *r;
})(5);
```

module - 模块

```
// 模块定义文件 circle.js
var PI = Math.PI; // 私有/局部

module.exports = function(r) {
  return {
    area: function() {
      return PI * r * r;
    },
    circumference: function() {
      return 2 * PI * r;
    }
  };
}

exports.getPi = function() {
  return PI;
}

// 使用文件 app.js
const circle = require('./circle.js');
console.log('圓周率: ' + circle.getPi());
```

module - 模块

- 加载策略
  - 部分原生模块有二进制执行文件，执行和加载速度都比较快
  - 静态加载和动态加载：require() 、runInThisContext()
  - 缓存机制
- require(‘xxx’)
  - 会根据后缀名决定加载方式：.js .node .json
  - 类型： 核心模块、文件模块、目录模块（index.js）、第三方模块（node_modules)


开始require
是否在文件模块缓存区中
是
否
是否原生模块
是
否
是否在原生模块缓存区中
否
加载原生模块
缓存原生模块
查找文件模块
根据扩展名载入文件模块
缓存文件模块
是
是
返回exports

package - 包

- 包结构
    - 一个package.json文件应该存在于包顶级目录下
    - 二进制文件应该包含在bin目录下
    - JavaScript代码应该包含在lib目录下
    - 文档应该在doc目录下
    - 单元测试应该在test目录下

创建一个工程

- 工具
  - npm init - 创建一个 package.json 文件
  - generator 工具： express-generator、angular-mongodb-generator …
  - 也可以手动生成：注意遵守一些默认的规则

DEMO