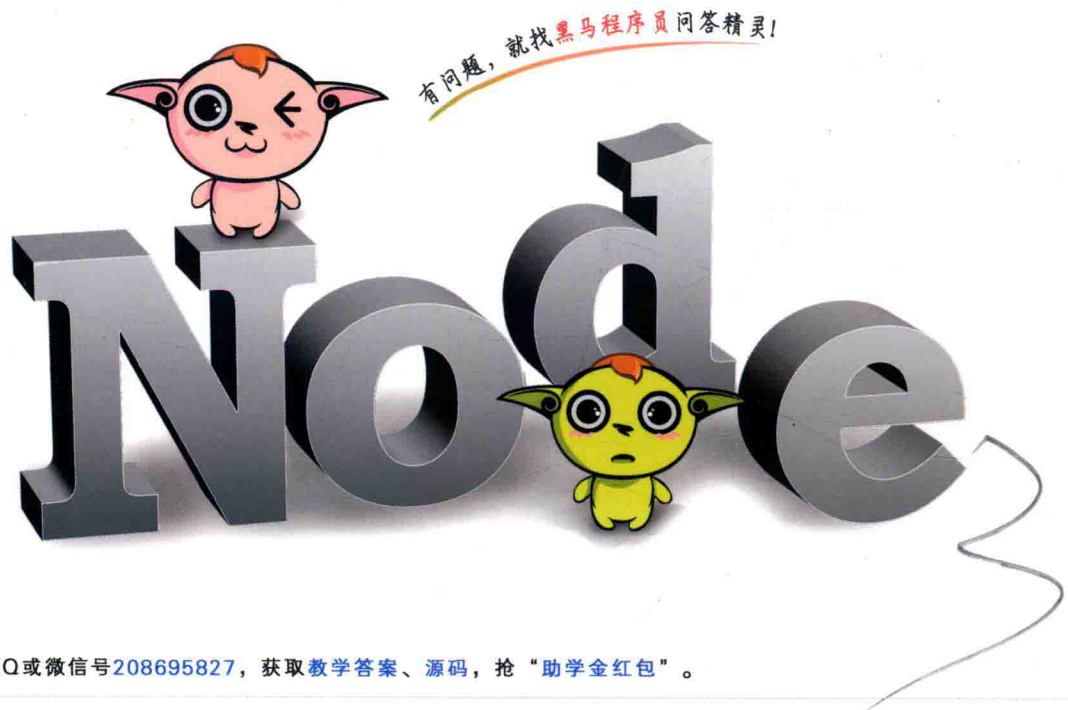
NITE 国家信息技术紧缺人才培养工程指定教材

教材+教案+授课资源+考试系统+题库+教学辅助案例

Node.js核心技术教程

Node.js HEXIN JISHU JIAOCHENG

黑马程序员 编著



添加QQ或微信号208695827，获取教学答案、源码，抢“助学金红包”。

采用案例与理论讲解相结合的教学方法，提供了41个案例，15道课后练习题。

提供免费教学资源，包括精美教学PPT、500道测试题、时长近20小时的教学视频等。

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

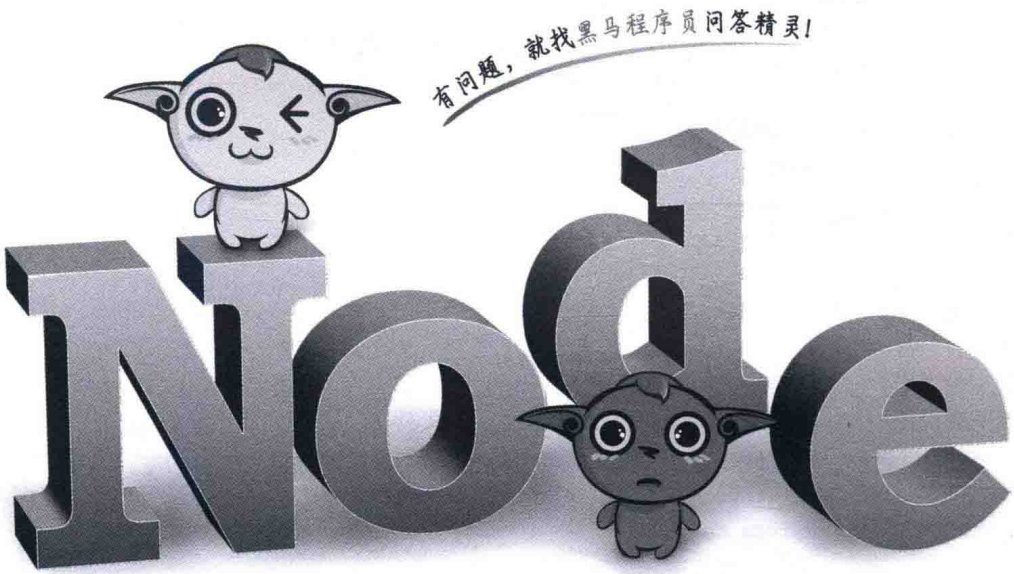
NITE 国家信息技术紧缺人才培养工程指定教材

教材+教案+授课资源+考试系统+题库+教学辅助案例

Node.js核心技术教程

Node.js HEXIN JISHU JIAOCHENG

黑马程序员 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

Node.js 是一个可以用 JavaScript 语言编写服务器端程序的开发平台。近几年，Node.js 逐渐发展为一个成熟的开发平台，吸引了许多编程人员，有许多大型网站都采用 Node.js 进行开发。本书详细讲解 Node.js 中的核心技术，包括模块化编程、异步编程、文件操作、数据处理、网络编程等内容。本书采用理论与操作相结合的方式进行讲解，以增加该技术的实用性和可操作性。在最后一章，综合前面的技术进行实际的项目编写，帮助读者学以致用。

本书适合作为高等院校计算机相关专业程序设计类课程或者 Web 开发的教材，也可作为广大计算机编程爱好者的参考用书。

图书在版编目（CIP）数据

Node. js核心技术教程 / 黑马程序员编著. —北京：中国铁道出版社，2017.4
国家信息技术紧缺人才培养工程指定教材
ISBN 978-7-113-22916-0

Ⅰ. ①N… Ⅱ. ①黑… Ⅲ. ①JAVA语言－程序设计－高等学校－教材 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字（2017）第050426号

书　　名：Node. js 核心技术教程
作　　者：黑马程序员　编著

策　　划：翟玉峰　　**读者热线**：（010）63550836
责任编辑：翟玉峰　彭立辉
封面设计：徐文海
封面制作：白　雪
责任校对：张玉华
责任印制：郭向伟

出版发行：中国铁道出版社（100054，北京市西城区右安门西街 8 号）
网　　址：http://www.tdpress.com/51eds/
印　　刷：中煤（北京）印务有限公司
版　　次：2017 年 4 月第 1 版　2017 年 4 月第 1 次印刷
开　　本：787 mm×1 092 mm　1/16　**印张**：11　**字数**：215 千
印　　数：1 ～ 3 000 册
书　　号：ISBN 978-7-113-22916-0
定　　价：32.00 元

播妞

播妞—— IT技术女神，由传智播客旗下高端教育品牌黑马程序员推出，专门服务于计算机相关专业的大学生及IT爱好者；可随时提供教材源代码、习题答案、免费视频教程和就业宝典等。

播妞倾情寄语：

你加，或者不加我
我就在这里
不离 不弃
来我这里
或者
让我住进你的心里

★ 小心！此处常有“助学金红包”出没！★
快来领取！

播妞QQ：208695827

播妞微信：208695827

教师获取教材配套资源

添加微信/QQ

2011168841

序

传智播客和“黑马程序员”

江苏传智播客教育科技股份有限公司（简称传智播客）是一家专门致力于高素质软件开发人才培养的科技公司，“黑马程序员”是传智播客旗下高端IT教育品牌。

“黑马程序员”的学员多为大学毕业后想从事IT行业，但各方面条件还不成熟的年轻人。“黑马程序员”的学员筛选制度非常严格，包括了严格的技术测试、自学能力测试、性格测试、压力测试、品德测试等。百里挑一的残酷筛选制度确保了学员质量，并降低了企业的用人风险。

自“黑马程序员”成立以来，教学研发团队一直致力于打造精品课程资源，不断在产、学、研3个层面创新自己的执教理念与教学方针，并集中“黑马程序员”的优势力量，有针对性地出版了计算机系列教材30多种，制作了教学视频数十套，发表各类技术文章数百篇。

“黑马程序员”不仅斥资研发IT系列教材，还为高校师生提供以下配套学习资源与服务。

为大学生提供的配套服务

（1）专注的辅学平台“博学谷”（http://yx.boxuegu.com），专业老师在线为您答疑解惑。

（2）针对高校学生在学习过程中存在的压力等问题，我们还面向大学生量身打造了“播妞”。播妞不仅致力推行快乐学习，还有定期的助学红包雨。同学快来添加播妞微信/QQ：208695827。

（3）高校学生也可扫描右方二维码，加入播妞粉丝团，获取最新学习资源，与播妞一起快乐学习。

为 IT 教师提供的配套服务

针对高校教学，“黑马程序员”为 IT 系列教材精心设计了“教案＋授课资源＋考试系统＋题库＋教学辅助案例”的系列教学资源，高校老师请关注码大牛老师微信 /QQ: 2011168841，获取教材配套资源，也可以扫描右方二维码，加入专为 IT 教师打造的师资服务平台——“教学好助手”，获取“黑马程序员”最新教师教学辅助资源相关动态。

黑马程序员

2017 年 2 月

前言

Node.js 发展及概要

随着互联网行业的持续发展，移动互联网等新业务不断发展壮大，相应的业务平台的开发形成了大量的人才缺口，尤其是 Web 前端。JavaScript 作为 Web 前端的核心技术，现在更是可以用于编写后台程序，这种进步就是由 Node.js 带来的。Node.js 是一个 JavaScript 运行环境，其优点为方便搭建、响应速度快、易于扩展等。Node.js 已成为 Web 前端编程人员必须掌握的一门新兴技术。

为什么要学习《Node.js 核心技术教程》

一个优秀的 Web 开发工程师需要具备一定的综合素质才能胜任企业日益复杂多变的要求，全栈工程师（Full Stack Engineer）的概念开始兴起。全栈工程师要熟练处理各层间的交互。Node.js 出现后，用 JavaScript 语言既可以进行客户端开发，又可以进行服务器端开发，还可以与数据库交互。这样便大大减少了开发人员的学习成本，为程序开发创造了良好的条件。本书正是讲解 Node.js 的核心技术。

如何使用本书

本书面向具有 JavaScript 基础的读者。请读者学习过 JavaScript 课程后，再学习本书。

本书详细讲解了 Node.js 的基本知识和使用方法，力求将一些非常复杂、难以理解的思想和问题简单化，让读者能够轻松理解并快速掌握。本书对每个知识点都进行了深入分析，并针对每个知识点精心设计了相关案例，帮助读者理解和掌握 Node.js 的核心技术，提高读者的实践操作能力。

本书共分为 8 章，下面分别对每个章节进行简要介绍。

（1）第 1 章主要介绍了模块化编程。Node.js 是一个高度模块化的平台，学习模块化思想可以帮助读者更好地理解和使用 Node.js。

（2）第 2 章讲解了 Node.js 的安装配置和一些基础概念。通过学习本章，读者已经准备好了开发环境和一些必备知识，为后面的核心技术奠定了基础。

（3）第 3 章讲解了异步编程和包资源管理。这也是 Node.js 中非常常见的操作。

（4）第 4 章主要讲解了 Node.js 文件操作。通过学习本章，读者可以很好地运用 Node.js 对文件进行读取、修改、复制等操作。

（5）第 5 章主要讲解了数据处理 I/O。数据与文件的处理是服务器端编程与客户端编程的本质区别所在，对于擅长前端编程的读者，应该重点掌握本章的内容，习惯用服务器端的思想来理解数据处理的问题。

（6）第 6 章主要讲解了网络编程的 Net 模块。通过学习本章，读者可以学会如何使用 Node.js 进行设备间数据的传输。

（7）第 7 章主要讲解网络编程的 HTTP 模块。通过学习本章，读者可以学会如何使用 Node.js 进行响应和请求的处理。

（8）第 8 章主要讲解了一个后台管理系统。通过学习本章，读者可以了解实际开发流程，实战用 Node.js 进行后台管理程序的编写。

在学习过程中，读者一定要亲自实践书中的案例代码。如果不能完全理解书中所讲知识，可以登录博学谷平台，通过平台中的教学视频进行深入学习。学习完一个知识点后，要及时在博学谷平台上进行测试，以巩固学习内容。另外，如果读者在理解知识点的过程中遇到困难，建议不要纠结于某个地方，可以先往后学习。通常来讲，看到后面对知识点的讲解或者其他小节的内容后，前面看不懂的知识点一般就能理解了。如果读者在动手练习的过程中遇到问题，建议多思考，理清思路，认真分析问题发生的原因，并在问题解决后多总结。

致谢

本书的编写和整理工作由传智播客教育科技股份有限公司完成，主要参与人员有吕春林、马丹、金鑫、马伦、刘晓强、汪磊等，全体人员在这近一年的编写过程中付出了很多辛勤的汗水，在此一并表示衷心的感谢。

意见反馈

尽管我们尽了最大的努力，但书中难免会有不妥之处，欢迎各界专家和读者朋友来信来函提出宝贵意见，我们将不胜感激。在阅读本书时，若发现任何问题或有不认同之处可以通过电子邮件与我们取得联系。

请发送电子邮件至 itcast_book@vip.sina.com。

黑马程序员

2017 年 2 月

目 录

第1章

模块化编程

网站开发中一些网页特效、数据处理等需要许多 JavaScript 代码进行支持，当开发越来越复杂时，会出现一些问题，例如命名冲突、烦琐的文件依赖等。为了解决这样的问题，JavaScript 模块化编程应运而生。本章将对 JavaScript 模块化编程进行详细讲解。

【教学导航】

学习目标	(1) 掌握模块化编程的概念 (2) 了解模块化编程的优势 (3) 了解模块化编程的演变 (4) 掌握常见的模块化解决方式
教学方式	以理论讲解、代码演示和案例效果展示为主
重点知识	(1) 理解模块化的设计思想 (2) 模块化程序的维护和扩展
关键词	模块化、全局函数、对象命名空间、函数的作用域

1.1 初识模块化思想

模块化是一种设计思想，利用模块化可以把一个非常复杂的系统结构细化到具体的功能点，每个功能点看作一个模块，然后通过某种规则把这些小的模块组合到一起，构成模块化系统。

1.1.1 模块化的概念

为了方便读者理解模块化的概念，先看一个现实生活中的模块化的例子，例如谷歌的模块化手机，如图 1-1 所示。

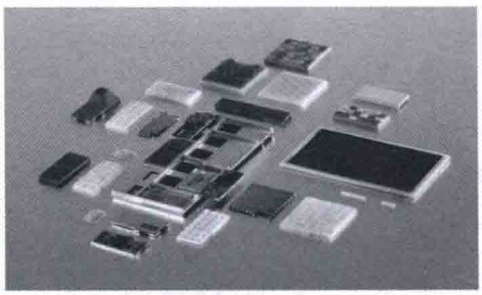


图 1-1 模块化手机

从图 1-1 可以看出，模块化手机分为多个模块，当某个模块损坏时都可以单独替换，也可以分模块进行手机升级。假如是一体机，某个部件损坏就要直接把手机换掉，这样的成本是不是很大？

从生产角度，模块化是一种生产方式，这种生产方式体现了两个特点：

(1) 生产效率高：

- 灵活架构，焦点分离。
- 多人协作互不干扰。
- 方便模块间组合、分解。

(2) 维护成本低：

- 可分单元测试。
- 方便单个模块功能调试、升级。

现在已经清楚了现实生活中的模块化，其实在程序中也有很多模块化的例子，例如程序中的常见日期模块（Date）、数学计算模块（Math）、日志模块、登录认证模块、报表展示模块等，所有模块组成一个程序软件系统。

同样，当某个模块出现问题时，只需要修改当前模块，而不影响其他模块的代码。程序模块化与现实生活中的模块化相似，从程序开发角度，模块化是一种开发模式，也有两个特点：

(1) 开发效率高：方便代码重用，对于别人开发好的模块功能可以直接拿过来使用，不需要重复开发类似的功能。

(2) 维护成本低：软件开发周期中，由于需求经常发生变化，最长的阶段并不是开发阶段，而是维护阶段，使用模块化开发的方式更容易维护。

1.1.2 模块化开发

了解了模块化后，读者可能有些疑问，模块化虽然有很多优势，但是它具体解决了编程人员在开发过程中的哪些问题？下面看一下非模块化开发会遇到哪些问题。

1. 命名冲突

在多人协作开发应用，或者使用第三方开发的 JavaScript 库的时候，通常会遇到命名冲突问题，例如全局变量中名称重复会报错，示例代码如下：

```
var  foo = 'bar';
var  foo = 'baz';
```

另外，如果引用第三方的 JavaScript 库，在全局对象中声明了一个属性 foo，自己的代码中也会声明同样名称的属性，两者一同使用的时候，后加载的属性值会替换之前的值，从而造成错误。

模块化开发的优点在于可以解决上述问题，让开发人员能很好地与他人协同，程序方面进行代码复用。那么，模块化是如何解决命名空间的呢？在 1.2 节中会有详细讲解。

2. 文件依赖

在开发过程中，可能需要很多文件依赖，示例代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>文件依赖</title>
</head>
<body>
  <script src="./ccc.js"></script>
  <script src="./a.js"></script>
  <script src="./b.js"></script>
  <script src="./c.js"></script>
  <script src="./d.js"></script>
  <script src="./aaa.js"></script>
</body>
</html>
```

从上述代码中，./aaa.js 是依赖于 ./a.js 文件的，但是从代码上并不能看出这样的关系。如果将 ./aaa.js 与 ./a.js 的前后位置调换，或者删除 ./a.js 文件，就会导致程序错误。

而在模块化开发中，会使用 JavaScript 代码来加载所需要的文件，并不需要将所有的文件引入到 HTML 文件中。

1.2 模块化编程的演变

使用非模块编程方式进行开发时，越大型的网站，命名冲突和文件依赖度越高，所以模块化编程对于大型网站来说具有重大的意义。那么，模块化编程是如何演变而来的呢？下面通过用不同的方式实现计算器的功能，一起来了解模块化编程的演变过程。

1.2.1 全局函数

首先从大多数人熟悉的编程习惯开始，假如现在要使用 JavaScript 实现一个计算器的案例，如图 1–2 所示。

图 1–2 计算器

在图 1–2 中，前两个文本框用于输入需要计算的数值，下拉菜单用于选择运算符，单击等号后，计算结果会出现在第三个文本框。

实现计算器第一种常见的写法是全局函数形式，示例代码如 demo1-1.html 所示。

demo1-1.html：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>模块化开发演变 - 全局函数</title>
</head>
<body>
  <input type="text" id="x">
  <select name="" id="opt">
    <option value="0">+</option>
    <option value="1">-</option>
    <option value="2">*</option>
    <option value="3">/</option>
  </select>
  <input type="text" id="y">
  <button id="cal">=</button>
  <input type="text" id="result">
  <script>
  // 定义用于计算的函数
    function add(x, y) {
      return parseInt(x) + parseInt(y);
    }

    function subtract(x, y) {
      return parseInt(x) - parseInt(y);
    }

    function multiply(x, y) {
      return parseInt(x) * parseInt(y);
    }

    function divide(x, y) {
```

```
33        return parseInt(x) / parseInt(y);
34      }
35    // 获取所有的 DOM 元素
36    var oX = document.getElementById('x');              // 第一个数值
37    var oY = document.getElementById('y');              // 第二个数值
38    var oOpt = document.getElementById('opt')           // 获取运算符
39    var oCal = document.getElementById('cal');          // 获取等号按钮
40    var oResult = document.getElementById('result')     // 结果数值
41    // 为等号按钮添加单击事件，当按钮被点击时调用此方法
42    oCal.addEventListener('click', function() {
43      var x = oX.value.trim()
44      var y = oY.value.trim()
45      var opt = oOpt.value
46      var result = 0
47      switch(opt) {
48        case '0':
49          result = add(x, y);                 // 加
50          break;
51        case '1':
52          result = subtract(x, y);            // 减
53          break;
54        case '2':
55          result = multiply(x, y);            // 乘
56          break;
57        case '3':
58          result = divide(x, y);              // 除
59          break;
60      }
61      oResult.value = result
62
63    })
64    </script>
65  </body>
66  </html>
```

在上述代码中，首先获取需计算的数值、运算符、等号按钮和结果数值的 DOM（文档对象模型）元素，然后分别定义了 4 个用来计算加、减、乘、除的函数，最后为等号按钮添加单击事件，通过 switch 语句判断调用哪个计算方法。

全局函数这种编程方式很常见，但是不可取，因为所有的变量和函数都暴露在全局，无法保证全局变量不与其他模块的变量发生冲突。另外，全局函数形成的模块成员之间看不出直接关系。