

JavaScript从入门到精通（标准版）



[JavaScript从入门到精通（标准版）_下载链接1](#)

著者:未来科技

出版者:中国水利水电出版社

出版时间:2017-7-1

装帧:平装

isbn:9787517054146

《JavaScript从入门到精通（标准版）》系统地讲解了JavaScript语言的使用，并结合HTML5介绍了如何开发更富可用性的Web程序。同属 JavaScript高级程序设计 JavaScript权威指南 JavaScript DOM 本JavaScript语言精粹 JavaScript算法 JavaScript入门 JavaScript实战 JavaScript设计模式大类。全书分为五大部分，共23章。第一部分介绍JavaScript的基本概念和基础知识，以及如何快速上手测试JavaScript代码。第二部分介绍JavaScript核心部分编程，包括变量、数据类型、表达式、运算符、语句、函数、数组、对象、正则表达式、函数式编程和面向对象编程。第三部分介绍了JavaScript客户端开发的相关知识和技术。第四部分讲JavaScript+HTML5的应用，如本地存储、图形绘制、定位、多线程、离线应用、文件操作等。第五部分通过多个综合实例演示了如何使用JavaScript进行实战开发的过程。

《JavaScript从入门到精通（标准版）》配备了极为丰富的学习资源，其中配套资源：290节教学视频（可二维码扫描）、素材源程序；附赠的拓展学习资源：习题及面试题库、案例库、工具库、网页模板库、网页配色库、网页素材库、网页案例欣赏库等。

《JavaScript从入门到精通（标准版）》适合作为JavaScript入门、JavaScript实战、JavaScript高级程序设计、HTML5移动开发方面的自学用书，也可作为高等院校网页设计、网页制作、网站建设、Web前端开发等专业的教学参考书或相关机构的培训教材。

作者介绍:

未来科技是由一群热爱Web开发的青年骨干教师组成的一个松散组织，主要从事Web开发、教学培训、教材开发等业务。该群体编写的同类图书在很多网店上的销量名列前茅，让数十万的读者轻松跨进了Web开发的大门，为Web开发的普及和应用做出了积极贡献。

目录: 第1章 JavaScript基础

1.1 JavaScript概述

1.1.1 JavaScript发展历史

1.1.2 ECMAScript与JavaScript的关系

1.1.3 ECMAScript版本变化

1.1.4 ECMAScript5和ECMAScript6

1.2 JavaScript相关概念

1.2.1 JavaScript核心

1.2.2 文档对象模型

1.2.3 浏览器对象模型

第2章 初次使用JavaScript

2.1 在网页中嵌入JavaScript脚本

2.1.1 编写脚本

2.1.2 脚本位置

2.1.3 设置延迟执行

2.1.4 设置异步响应

2.2 执行JavaScript程序

2.2.1 执行过程

2.2.2 预编译

2.2.3 代码块

2.2.4 响应事件

2.2.5 设计动态脚本

第3章 代码测试和错误处理

3.1 浏览器与JavaScript

3.1.1 浏览器内核

3.1.2 浏览器错误报告

3.2 JavaScript开发工具

3.2.1 JavaScript编辑器

3.2.2 JavaScript测试和调试

3.2.3 使用控制台

3.3 错误处理

3.3.1 认识错误类型

3.3.2 使用try-catch

3.3.3 使用finally

3.3.4 使用throw

3.3.5 抛出时机

3.3.6 错误事件

第4章 JavaScript基本语法

4.1 基本词法

4.1.1 字符编码

4.1.2 区分大小写

4.1.3 标识符

4.1.4 直接量

4.1.5 关键字和保留字

4.1.6 分隔符

4.1.7 注释

4.1.8 转义序列

4.2 使用变量

- 4.2.1 声明变量
- 4.2.2 赋值变量
- 4.2.3 变量的作用域
- 4.2.4 避免变量污染
- 4.3 数据类型
 - 4.3.1 基本数据类型
 - 4.3.2 数值
 - 4.3.3 字符串
 - 4.3.4 布尔值
 - 4.3.5 Null
 - 4.3.6 Undefined
- 4.4 严格模式
 - 4.4.1 启用严格模式
 - 4.4.2 严格模式的执行限制
- 4.5 案例实战
 - 4.5.1 使用typeof检测类型
 - 4.5.2 使用constructor检测类型
 - 4.5.3 封装类型检测方法：toString()

- 4.5.4 转换为字符串
- 4.5.5 转换数字模式
- 4.5.6 设置数字显示的小数位数
- 4.5.7 转换为数字
- 4.5.8 转换为布尔值
- 4.5.9 转换为对象
- 4.5.10 把对象转换为值
- 4.5.11 强制转换

第5章 使用运算符

- 5.1 运算符概述
- 5.2 算术运算符
 - 5.2.1 加法运算
 - 5.2.2 减法运算
 - 5.2.3 乘法运算
 - 5.2.4 除法运算
 - 5.2.5 余数运算
 - 5.2.6 取反运算
 - 5.2.7 递增和递减
- 5.3 逻辑运算符
 - 5.3.1 逻辑与运算
 - 5.3.2 逻辑或运算符
 - 5.3.3 逻辑非运算符
 - 5.3.4 案例：逻辑运算训练
- 5.4 关系运算符
 - 5.4.1 大小比较
 - 5.4.2 案例：包含检测
 - 5.4.3 案例：等值检测
- 5.5 赋值运算符
- 5.6 对象操作运算符
 - 5.6.1 new运算符
 - 5.6.2 delete运算符
 - 5.6.3 中括号和点号运算符
 - 5.6.4 小括号运算符
- 5.7 其他运算符
 - 5.7.1 条件运算符

- 5.7.2 逗号运算符
- 5.7.3 void运算符
- 5.8 案例实战
 - 5.8.1 使用表达式
 - 5.8.2 连续运算
 - 5.8.3 把命令转换为表达式
 - 5.8.4 表达式中的函数
- 第6章 设计程序结构
 - 6.1 语句概述
 - 6.1.1 表达式语句
 - 6.1.2 复合语句
 - 6.1.3 声明语句
 - 6.1.4 空语句
 - 6.2 分支结构
 - 6.2.1 if语句
 - 6.2.2 条件嵌套
 - 6.2.3 设计分支结构
 - 6.2.4 switch语句
 - 6.2.5 default从句
 - 6.2.6 比较if和switch
 - 6.2.7 优化分支结构
 - 6.3 循环结构
 - 6.3.1 while语句
 - 6.3.2 do/while语句
 - 6.3.3 for语句
 - 6.3.4 for/in语句
 - 6.3.5 比较while和for
 - 6.3.6 优化循环结构
 - 6.4 结构跳转
 - 6.4.1 标签语句
 - 6.4.2 break语句
 - 6.4.3 continue语句
 - 6.5 案例实战
 - 6.5.1 提升分支运算性能
 - 6.5.2 提升循环运算性能
 - 6.5.3 设计杨辉三角
- 第7章 使用数组
 - 7.1 定义数组
 - 7.1.1 构造数组
 - 7.1.2 数组直接量
 - 7.2 使用数组
 - 7.2.1 存取数组元素
 - 7.2.2 数组长度
 - 7.2.3 对象与数组
 - 7.2.4 定义多维数组
 - 7.3 使用数组对象
 - 7.3.1 检索数组
 - 7.3.2 操作元素
 - 7.3.3 操作子数组
 - 7.3.4 数组排序
 - 7.3.5 使用排序函数
 - 7.3.6 数组与字符串的转换
 - 7.3.7 定位
 - 7.3.8 迭代

- 7.3.9 汇总
- 7.4 案例实战
 - 7.4.1 快速交换
 - 7.4.2 数组下标
 - 7.4.3 扩展数组方法
 - 7.4.4 设计迭代器
 - 7.4.5 使用迭代器
 - 7.4.6 使用数组维度
- 第8章 使用函数
 - 8.1 定义函数
 - 8.1.1 声明函数
 - 8.1.2 构造函数
 - 8.1.3 函数直接量
 - 8.1.4 定义嵌套函数
 - 8.1.5 比较定义函数的方法
 - 8.2 使用函数
 - 8.2.1 函数返回值
 - 8.2.2 调用函数
 - 8.2.3 函数作用域
 - 8.3 使用参数
 - 8.3.1 定义参数
 - 8.3.2 使用arguments对象
 - 8.3.3 使用callee回调函数
 - 8.3.4 应用arguments对象
 - 8.4 使用函数对象
 - 8.4.1 获取函数形参个数
 - 8.4.2 自定义属性
 - 8.4.3 使用call()和apply()
 - 8.4.4 使用bind()
 - 8.5 使用this
 - 8.5.1 使用this
 - 8.5.2 this安全策略
 - 8.5.3 应用this
 - 8.5.4 函数调用模式
 - 8.5.5 函数的标识符
 - 8.6 使用闭包函数
 - 8.6.1 认识闭包函数
 - 8.6.2 使用闭包
 - 8.6.3 定义闭包存储器
 - 8.6.4 在事件处理中应用闭包
 - 8.7 案例实战
 - 8.7.1 绑定函数
 - 8.7.2 链式语法
 - 8.7.3 函数节流
 - 8.7.4 分支函数
 - 8.7.5 惰性载入函数
 - 8.7.6 惰性求值
 - 8.7.7 记忆
 - 8.7.8 构建模块
 - 8.7.9 柯里化
 - 8.7.10 高阶函数
 - 8.7.11 递归运算
 - 8.7.12 尾递归算法
- 第9章 使用对象

- 9.1 创建对象
 - 9.1.1 使用构造函数创建对象
 - 9.1.2 使用对象直接量创建对象
 - 9.1.3 使用create()方法创建对象
- 9.2 操作对象
 - 9.2.1 引用对象
 - 9.2.2 复制对象
 - 9.2.3 克隆继承
 - 9.2.4 销毁对象
- 9.3 操作属性
 - 9.3.1 定义属性
 - 9.3.2 访问属性
 - 9.3.3 赋值属性
 - 9.3.4 删除属性
 - 9.3.5 使用方法
 - 9.3.6 配置特性
 - 9.3.7 检测特性
- 9.4 使用方法
 - 9.4.1 使用toString()
 - 9.4.2 使用valueOf()
 - 9.4.3 检测私有属性
 - 9.4.4 检测枚举属性
 - 9.4.5 检测原型对象
 - 9.4.6 静态方法
- 9.5 使用原型
 - 9.5.1 定义原型
 - 9.5.2 比较原型属性和本地属性
 - 9.5.3 应用原型
 - 9.5.4 原型域和原型域链
 - 9.5.5 原型继承
 - 9.5.6 扩展原型方法
- 9.6 案例实战
 - 9.6.1 设计工厂模式
 - 9.6.2 设计类继承
 - 9.6.3 设计构造原型模式
 - 9.6.4 设计动态原型模式
 - 9.6.5 设计实例继承
 - 9.6.6 惰性实例化
 - 9.6.7 安全构造对象
- 第10章 BOM操作
 - 10.1 使用window对象
 - 10.1.1 访问浏览器窗口
 - 10.1.2 全局作用域
 - 10.1.3 使用系统测试方法
 - 10.1.4 打开和关闭窗口
 - 10.1.5 使用框架集
 - 10.1.6 控制窗口位置
 - 10.1.7 控制窗口大小
 - 10.1.8 使用定时器
 - 10.2 使用navigator对象
 - 10.2.1 浏览器检测方法
 - 10.2.2 检测浏览器类型和版本号
 - 10.2.3 检测客户操作系统
 - 10.2.4 检测插件

- 10.3 使用location对象
- 10.4 使用history对象
- 10.5 使用screen对象
- 10.6 使用document对象
- 10.6.1 访问文档对象
- 10.6.2 动态生成文档内容
- 10.7 案例实战
- 10.7.1 使用远程脚本
- 10.7.2 设计远程交互
- 10.7.3 使用浮动框架
- 10.7.4 封装用户代理检测
- 第11章 DOM操作
- 11.1 DOM基础
- 11.2 使用节点
- 11.2.1 节点类型
- 11.2.2 节点名称和值
- 11.2.3 节点关系
- 11.2.4 访问节点
- 11.2.5 操作节点
- 11.3 使用文档节点
- 11.3.1 访问文档子节点
- 11.3.2 访问文档信息
- 11.3.3 访问文档元素
- 11.3.4 访问文档集合
- 11.3.5 使用HTML5 Document
- 11.4 使用元素节点
- 11.4.1 访问元素
- 11.4.2 遍历元素
- 11.4.3 创建元素
- 11.4.4 复制节点
- 11.4.5 插入节点
- 11.4.6 删除节点
- 11.4.7 替换节点
- 11.4.8 获取焦点元素
- 11.4.9 检测包含节点
- 11.5 使用文本节点
- 11.5.1 访问文本节点
- 11.5.2 创建文本节点
- 11.5.3 操作文本节点
- 11.5.4 读取HTML字符串
- 11.5.5 插入HTML字符串
- 11.5.6 替换HTML字符串
- 11.5.7 插入文本
- 11.6 使用文档片段节点
- 11.7 使用属性节点
- 11.7.1 访问属性节点
- 11.7.2 读取属性值
- 11.7.3 设置属性值
- 11.7.4 删除属性
- 11.7.5 使用类选择器
- 11.7.6 自定义属性
- 11.8 使用范围
- 11.8.1 创建范围
- 11.8.2 选择范围

- 11.8.3 设置范围
- 11.8.4 操作范围内容
- 11.8.5 插入范围内容
- 11.8.6 折叠范围
- 11.8.7 比较范围
- 11.8.8 复制和清除范围
- 11.9 使用CSS选择器
- 11.10 案例实战
 - 11.10.1 设计动态脚本
 - 11.10.2 使用script加载远程数据
 - 11.10.3 使用script实现异步交互
 - 11.10.4 使用JSONP
 - 11.10.5 设计动态表格
 - 11.10.6 访问DOM集合
 - 11.10.7 在微博分享选中文本
- 第12章 事件处理
 - 12.1 事件基础
 - 12.1.1 事件模型
 - 12.1.2 事件流
 - 12.1.3 事件类型
 - 12.1.4 绑定事件
 - 12.1.5 事件处理函数
 - 12.1.6 注册事件
 - 12.1.7 销毁事件
 - 12.1.8 使用event对象
 - 12.1.9 事件委托
 - 12.2 使用鼠标事件
 - 12.2.1 鼠标点击
 - 12.2.2 鼠标移动
 - 12.2.3 鼠标经过
 - 12.2.4 鼠标来源
 - 12.2.5 鼠标定位
 - 12.2.6 鼠标按键
 - 12.3 使用键盘事件
 - 12.3.1 键盘事件属性
 - 12.3.2 键盘响应顺序
 - 12.4 使用页面事件
 - 12.4.1 页面初始化
 - 12.4.2 结构初始化
 - 12.4.3 页面卸载
 - 12.4.4 窗口重置
 - 12.4.5 页面滚动
 - 12.4.6 错误处理
 - 12.5 使用UI事件
 - 12.5.1 焦点处理
 - 12.5.2 选择文本
 - 12.5.3 字段值变化监测
 - 12.5.4 提交表单
 - 12.5.5 重置表单
 - 12.5.6 剪贴板数据
 - 12.6 案例实战
 - 12.6.1 封装事件
 - 12.6.2 模拟事情
 - 12.6.3 设计弹出对话框

- 12.6.4 设计遮罩层
- 12.6.5 自定义事件
- 12.6.6 设计事件触发模型
- 12.6.7 应用事件模型
- 第13章 使用正则表达式与表单验证
- 13.1 正则表达式操作基础
 - 13.1.1 定义正则表达式
 - 13.1.2 访问正则表达式对象
 - 13.1.3 执行匹配操作
 - 13.1.4 访问匹配信息
 - 13.1.5 条件检测
- 13.2 正则表达式语法基础
 - 13.2.1 字符描述
 - 13.2.2 字符范围
 - 13.2.3 选择操作
 - 13.2.4 重复类量词
 - 13.2.5 惰性模式
 - 13.2.6 边界量词
 - 13.2.7 声明量词
 - 13.2.8 表达式分组
 - 13.2.9 子表达式引用
- 13.3 案例实战
- 第14章 字符串处理与表单开发
- 14.1 字符串操作基础
 - 14.1.1 定义字符串
 - 14.1.2 字符串的值和字符长度
 - 14.1.3 字符串连接
 - 14.1.4 字符串查找
 - 14.1.5 字符串截取
 - 14.1.6 字符串替换
 - 14.1.7 字符串大小转换
 - 14.1.8 字符串比较
 - 14.1.9 字符串与数组转换
 - 14.1.10 字符串格式化
 - 14.1.11 字符编码和解码
 - 14.1.12 Unicode编码和解码
- 14.2 案例实战
 - 14.2.1 访问表单对象
 - 14.2.2 访问表单元素
 - 14.2.3 访问字段属性
 - 14.2.4 访问文本框的值
 - 14.2.5 文本框过滤
 - 14.2.6 切换焦点
 - 14.2.7 访问选择框的值
 - 14.2.8 编辑选项
 - 14.2.9 字符串替换的高级应用
 - 14.2.10 字符串修剪
 - 14.2.11 检测特殊字符
 - 14.2.12 自定义加密和解密
 - 14.2.13 表单序列化
 - 14.2.14 设计文本编辑器
- 第15章 CSS脚本化与网页特效
- 15.1 CSS脚本化基础
 - 15.1.1 访问CSS行内样式

- 15.1.2 使用style对象
- 15.1.3 使用styleSheets对象
- 15.1.4 使用selectorText对象
- 15.1.5 编辑样式
- 15.1.6 添加样式
- 15.1.7 访问计算样式
- 15.2 元素大小
 - 15.2.1 访问CSS宽度和高度
 - 15.2.2 把值转换为整数
 - 15.2.3 使用offsetWidth和offset Height
 - 15.2.4 元素尺寸
 - 15.2.5 视图尺寸
 - 15.2.6 窗口尺寸
- 15.3 位置偏移
 - 15.3.1 窗口位置
 - 15.3.2 相对位置
 - 15.3.3 定位位置
 - 15.3.4 设置偏移位置
 - 15.3.5 设置相对位置
 - 15.3.6 鼠标指针绝对位置
 - 15.3.7 鼠标指针相对位置
 - 15.3.8 滚动条位置
 - 15.3.9 设置滚动条位置
- 15.4 显示隐藏
 - 15.4.1 可见性
 - 15.4.2 透明度
- 15.5 案例实战
 - 15.5.1 滑动
 - 15.5.2 渐显
- 第16章 使用Ajax实现异步通信
 - 16.1 使用XML数据
 - 16.1.1 新建XML文档
 - 16.1.2 访问XML数据
 - 16.1.3 创建XML DOM对象
 - 16.1.4 加载XML数据
 - 16.1.5 显示XML数据
 - 16.1.6 案例：在网页中显示XML数据
 - 16.1.7 案例：异步加载XML数据
 - 16.2 使用JSON数据
 - 16.2.1 JSON结构
 - 16.2.2 案例：JSON与XML格式比较
 - 16.2.3 案例：JSON数据优化
 - 16.2.4 案例：解析JSON
 - 16.2.5 案例：序列化JSON
 - 16.3 使用Ajax
 - 16.3.1 HTTP头部信息
 - 16.3.2 定义XMLHttpRequest对象
 - 16.3.3 建立XMLHttpRequest连接
 - 16.3.4 发送GET请求
 - 16.3.5 发送POST请求
 - 16.3.6 转换串行化字符串
 - 16.3.7 跟踪状态
 - 16.3.8 中止请求
 - 16.3.9 获取XML数据

- 16.3.10 获取HTML文本
- 16.3.11 获取JavaScript脚本
- 16.3.12 获取JSON数据
- 16.3.13 获取纯文本
- 16.3.14 获取头部信息
- 16.4 案例实战
 - 16.4.1 封装异步请求操作
 - 16.4.2 动态显示提示信息
 - 16.4.3 动态查询记录集
 - 16.4.4 记录集分页显示
 - 16.4.5 设计Tab面板
 - 16.4.6 关键字匹配
 - 16.4.7 使用灯标
- 第17章 本地数据存储571
 - 17.1 使用cookie
 - 17.1.1 写入cookie信息
 - 17.1.2 读取cookie信息
 - 17.1.3 修改和删除cookie信息
 - 17.1.4 附加cookie信息
 - 17.1.5 封装cookie操作
 - 17.1.6 案例：打字游戏
 - 17.2 使用Web Storage
 - 17.2.1 基本操作
 - 17.2.2 案例：设计网页皮肤
 - 17.2.3 案例：跟踪localStorage数据
 - 17.2.4 案例：设计计数器
 - 17.3 使用Web SQL
 - 17.3.1 基本操作
 - 17.3.2 案例：创建本地数据库
 - 17.3.3 案例：批量存储本地数据
 - 17.4 案例实战
- 第18章 JavaScript图形设计
 - 18.1 HTML 5 canvas基础
 - 18.1.1 在页面中插入canvas元素
 - 18.1.2 绘制图形的基本方法
 - 18.1.3 使用canvas
 - 18.2 绘制图形
 - 18.2.1 绘制直线
 - 18.2.2 绘制矩形
 - 18.2.3 绘制圆形
 - 18.2.4 绘制多边形
 - 18.2.5 绘制曲线
 - 18.2.6 绘制二次方曲线
 - 18.2.7 绘制三次方曲线
 - 18.3 设置图形样式
 - 18.3.1 设置线型
 - 18.3.2 绘制线性渐变
 - 18.3.3 绘制径向渐变
 - 18.3.4 绘制图案
 - 18.3.5 设置不透明度
 - 18.3.6 设置阴影
 - 18.4 操作图形
 - 18.4.1 保存和恢复canvas状态
 - 18.4.2 清除绘图

- 18.4.3 移动坐标
- 18.4.4 旋转坐标
- 18.4.5 缩放图形
- 18.4.6 变换矩阵
- 18.4.7 组合图形
- 18.4.8 裁切路径
- 18.5 绘制文字
 - 18.5.1 绘制填充文字
 - 18.5.2 设置文字属性
 - 18.5.3 绘制轮廓文字
 - 18.5.4 测量宽度
- 18.6 绘制图像
 - 18.6.1 导入图像
 - 18.6.2 变换图像
 - 18.6.3 裁切图像
 - 18.6.4 图像平铺
 - 18.6.5 像素处理
- 18.7 案例实战
 - 18.7.1 设计canvas动画
 - 18.7.2 保存绘图
- 第19章 离线应用
 - 19.1 HTML5离线应用基础
 - 19.1.1 认识HTML5离线应用
 - 19.1.2 浏览器支持
 - 19.1.3 使用manifest文件
 - 19.1.4 使用离线缓存
 - 19.1.5 监听离线存储
 - 19.2 案例实战
 - 19.2.1 缓存首页
 - 19.2.2 离线编辑内容
 - 19.2.3 离线跟踪
- 第20章 多线程处理
 - 20.1 Web Workers基础
 - 20.1.1 认识Web Workers
 - 20.1.2 浏览器支持
 - 20.1.3 创建Web Workers
 - 20.1.4 Web Workers通信674
 - 20.1.5 案例：使用Web Workers675
 - 20.2 案例实战
 - 20.2.1 后台运算
 - 20.2.2 数值过滤
 - 20.2.3 并发处理
 - 20.2.4 线程通信
 - 20.2.5 Fibonacci数列运算
- 第21章 文件操作
 - 21.1 访问文件域
 - 21.2 使用Blob对象
 - 21.2.1 在文件域中访问Blob对象
 - 21.2.2 创建Blob对象
 - 21.2.3 截取Blob对象
 - 21.2.4 保存Blob对象
 - 21.3 使用FileReader对象
 - 21.3.1 读取并显示文件
 - 21.3.2 监测读取操作

- 21.4 使用缓存对象
 - 21.4.1 使用ArrayBuffer对象
 - 21.4.2 使用ArrayBufferView对象
 - 21.4.3 使用DataView对象
- 21.5 使用FileSystem
 - 21.5.1 访问文件系统
 - 21.5.2 申请配额
 - 21.5.3 创建文件
 - 21.5.4 写入文件
 - 21.5.5 添加数据
 - 21.5.6 读取文件
 - 21.5.7 复制文件
 - 21.5.8 删除文件
 - 21.5.9 创建目录
 - 21.5.10 读取目录
 - 21.5.11 删除目录
 - 21.5.12 复制目录
 - 21.5.13 移动和重命名目录
 - 21.5.14 使用filesystem:URL
- 21.6 案例实战
- 第22章 使用History
 - 22.1 History API基础
 - 22.1.1 History API处理方式
 - 22.1.2 浏览器兼容和扩展
 - 22.1.3 操作历史记录
 - 22.2 案例实战
 - 22.2.1 设计无刷新页面导航
 - 22.2.2 设计主题宣传网站
 - 22.2.3 设计图片画廊
 - 22.2.4 设计历史恢复
- 第23章 案例实战
 - 23.1 设计折叠面板
 - 23.2 设计计算器
 - 23.3 设计万年历
 - 23.4 设计俄罗斯方块
 - 23.4.1 设计游戏界面
 - 23.4.2 设计游戏模型
 - 23.4.3 实现游戏功能
 - • • • • [\(收起\)](#)

[JavaScript从入门到精通（标准版）](#) [下载链接1](#)

标签

科-编程

研发

评论

[JavaScript从入门到精通（标准版）_下载链接1](#)

书评

[JavaScript从入门到精通（标准版）_下载链接1](#)