

HTML5

交互动画 开发实践教程

杜晓荣 主 编 |
徐泳钧 武汇岳 副主编



清华大学出版社

HTML5

交互动画 开发实践教学

徐泳钧 杜晓荣 主 编 |
武汇岳 副主编

清华大学出版社
北 京

内 容 简 介

本书分3部分,第一部分是JavaScript脚本语言的基本教程,在该部分集中讲述了用于web网页开发的脚本语言JavaScript,用简明通俗的语言介绍了JavaScript的面向对象技术;第二部分讲解了HTML5画布的基本绘图、高级绘图及基于引擎的绘图功能;第三部分则以动画的例程和休闲游戏例程来讲述如何利用原生的HTML5画布功能来设计动画与休闲游戏的开发。通过本书系统的学习,读者可以掌握JavaScript的面向对象技术、HTML5的图形功能、动画和休闲游戏的开发及编程技术。

本书适用于希望学习HTML5新技术以及Web前端开发人员,也可用作高校数字媒体专业、动画设计专业或网页开发相关专业的教材。教材面向所有对动态网页和交互动画感兴趣的读者,授课内容将由浅入深,理论和实践相结合,从基本语法逐渐过渡到读者亲自设计动画交互,激发读者对网页交互设计的兴趣。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

HTML5交互动画开发实践教程/杜晓荣主编. —北京:清华大学出版社,2014

ISBN 978-7-302-37605-7

I. ①H… II. ①杜… III. ①超文本标记语言—程序—设计—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第186544号

责任编辑:朱敏悦

封面设计:汉风唐韵

责任校对:王荣静

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:16.5

字 数:338千字

版 次:2014年9月第1版

印 次:2014年9月第1次印刷

印 数:1~4000

定 价:33.00元

产品编号:061459-01

前 言

本书适用于希望学习 HTML5 新技术以及 Web 前端开发人员,也可用于高校数字媒体专业、动画设计专业或相关专业的教材。本书不要求读者具有编程经验,不过如果具有编程基础,那么会对本书内容更容易了解。通过本书系统的学习,读者可以掌握 JavaScript 的面向对象技术、HTML5 的图形功能、动画和休闲游戏的编程及开发技术。

本书共分为 3 个部分,第一部分是 JavaScript 脚本语言的基本教程,在该部分集中讲述了用于 Web 网页开发的脚本语言 JavaScript,用简明通俗的语言介绍了 JavaScript 的面向对象技术;第二部分讲解了 HTML5 画布的基本绘图、高级绘图及基于引擎的绘图功能;第三部分则以动画的例程和休闲游戏例程来讲述如何利用原生的 HTML5 画布功能设计动画与休闲游戏的开发。

第一部分

第 1 章 基本概念 本章简单介绍了 HTML5 的发展现状,并且演示了搭建开发环境的步骤流程。

第 2 章 编程基础 本章开始介绍 JavaScript 的基本语法,包括变量命名、数据类型与运算符的作用,编程的书写规范。读者可以了解到 JavaScript 与 C 语言风格的相似之处与不同点,并为之后的学习打好基础。

第 3 章 基本流程控制 本章中讲述了 JavaScript 用于流程控制的基本语句,包括顺序语句、条件语句和判断语句。

第 4 章 函数 本章中,读者可以学习到函数的定义方法,函数参数和返回值的相关知识,读者可以尝试编写一个函数并在语句中调用这个函数。

第 5 章 引用类型 本章介绍了 JavaScript 中的核心类型,即引用类型。JavaScript 是一种基于对象的语言,因此在 JavaScript 中非常多的概念都是通过一个对象来实现,甚至连函数也是一个对象。读者可以从本章学习到 JavaScript 中的内置对象以及访问对象的属性和方法。

第 6 章 面向对象编程 本章中介绍了如何在 JavaScript 中编写面向对象编程的代码,包括对象的封装、继承和多态的实现。

第二部分

第 7 章 Canvas 基本功能 本章中介绍了画布提供的基本绘图功能,读者可以在这



章中学到如何利用画布提供的方法来实现画布上下文中绘图,以及改变绘图属性来绘制不同样式的图案。

第8章 Canvas 高级功能 本章在第7章的基础上,更深一层地讲述了画布提供的高级绘图功能。利用本章中的内容,读者可以绘制出带有更加复杂效果的图案。

第9章 CVIDrawJS 绘图部分 本章介绍了中山大学自主研制的 CVIDrawJS 游戏引擎绘图部分的功能。(通过面向对象编程把绘图方法封装在对象之中,)使得开发者可以方便地调用绘图对象的方法来快速绘图。

第三部分

第10章 预备知识 本章中介绍了动画的形成过程和浏览器上的设备响应的实现。本章主要是为第11、12章的内容做基础。读者可以在本章中学习到矩阵变换所形成的动画和精灵动画,以及与浏览器进行交互的方法。

第11章 HTML5 动画设计 本章以一个鱼游动动画的设计为主线,一步步介绍了如何使用 HTML5 的原生接口进行动画的设计与制作。读者可以学到简单动画的设计过程。

第12章 HTML5 休闲游戏设计 本章介绍了浏览器上休闲小游戏的制作过程。其中游戏制作的过程从简单的游戏原型开始,一步步增添功能和完善游戏,带领读者了解一个简单的休闲游戏制作的全过程。

目 录

第一部分

第 1 章 基本概念	(3)
1.1 了解 HTML5	(3)
1.1.1 什么是 HTML5	(3)
1.1.2 HTML5 新特性	(3)
1.1.3 HTML5 发展趋势	(4)
1.2 了解 JavaScript	(4)
1.2.1 什么是 JavaScript	(4)
1.2.2 核心(ECMAScript)	(5)
1.2.3 文档对象模型(DOM)	(5)
1.2.4 浏览器对象模型(BOM)	(7)
1.3 搭建开发环境	(7)
1.3.1 开发环境介绍	(8)
1.3.2 代码编辑器	(8)
1.3.3 浏览器	(12)
1.4 在 HTML 文档中使用 JavaScript	(14)
1.4.1 新建 HTML 文档	(14)
1.4.2 直接嵌入 JavaScript	(15)
1.4.3 嵌入 JavaScript 脚本文件	(17)
1.5 小结	(18)
1.6 习题	(18)
第 2 章 编程基础	(20)
2.1 编程规范	(20)
2.1.1 注释	(20)



2.1.2 命名规范	(20)
2.2 变量	(21)
2.2.1 变量命名	(21)
2.2.2 变量声明和赋值	(22)
2.2.3 变量的作用域	(22)
2.3 数据类型	(23)
2.3.1 Undefined 类型	(24)
2.3.2 Null 类型	(24)
2.3.3 Number 类型	(24)
2.3.4 Boolean 类型	(29)
2.3.5 String 类型	(29)
2.3.6 Object 类型	(30)
2.4 运算符与表达式	(31)
2.4.1 表达式的含义	(31)
2.4.2 JavaScript 中的运算符	(31)
2.4.3 运算符的优先级	(32)
2.4.4 运算符的结合性	(33)
2.4.5 算术运算符	(33)
2.4.6 关系运算符	(37)
2.4.7 逻辑运算符	(39)
2.4.8 条件运算符	(41)
2.4.9 位操作运算符	(42)
2.4.10 其他运算符	(46)
2.5 小结	(48)
2.6 习题	(49)
第3章 基本流程控制	(50)
3.1 if 语句	(50)
3.2 switch 语句	(54)
3.3 while 语句	(57)
3.4 do-while 语句	(58)
3.5 for 语句	(59)
3.6 for-in 语句	(60)
3.7 break 和 continue 语句	(61)

3.8 小结	(63)
3.9 习题	(64)
第4章 函数	(65)
4.1 什么是函数	(65)
4.2 函数定义	(66)
4.3 函数参数	(67)
4.3.1 形参和实参	(67)
4.3.2 形参和实参数目	(68)
4.3.3 arguments 对象	(70)
4.3.4 模拟函数重载	(71)
4.4 函数调用返回值	(72)
4.5 递归函数	(73)
4.6 小结	(74)
4.7 习题	(75)
第5章 引用类型	(76)
5.1 基本类型和引用类型	(76)
5.1.1 内存保存	(76)
5.1.2 复制变量值	(77)
5.2 引用类型	(79)
5.2.1 Object 类型	(79)
5.2.2 Array 类型	(81)
5.2.3 Function 类型	(85)
5.3 基本数据类型的方法	(87)
5.3.1 Boolean 类型	(87)
5.3.2 Number 类型	(88)
5.3.3 String 类型	(89)
5.4 Math 对象	(91)
5.5 小结	(93)
5.6 习题	(93)
第6章 面向对象编程	(96)
6.1 概述	(96)
6.2 this 对象	(97)
6.3 封装	(98)



6.3.1	构造函数	(99)
6.3.2	原型方法	(101)
6.3.3	混合方式	(103)
6.4	继承	(106)
6.4.1	对象冒充	(106)
6.4.2	call()和 apply()方法	(107)
6.4.3	原型链	(109)
6.4.4	混合方式继承	(111)
6.5	多态性实现	(112)
6.5.1	重写父对象方法	(112)
6.5.2	多态性	(113)
6.6	小结	(115)
6.7	习题	(115)

第二部分

第7章	Canvas 基本功能	(119)
7.1	<canvas>元素	(119)
7.1.1	引入<canvas>元素	(120)
7.1.2	获取上下文	(120)
7.2	绘制简单图形	(121)
7.2.1	绘制直线	(121)
7.2.2	线条属性	(123)
7.2.3	闭合图形	(126)
7.2.4	绘制矩形	(127)
7.2.5	绘制圆弧	(129)
7.2.6	贝塞尔曲线	(132)
7.3	绘制文本	(133)
7.3.1	绘制文本	(133)
7.3.2	文本属性	(135)
7.3.3	测量文本宽度	(139)
7.4	小结	(140)
7.5	习题	(140)

第 8 章 Canvas 高级功能	(142)
8.1 绘制图像	(142)
8.2 像素级操作	(144)
3.2.1 像素对象	(144)
8.2.2 创建像素对象	(145)
8.2.3 取得像素对象	(146)
8.2.4 绘制像素对象	(147)
8.3 变换	(148)
8.3.1 平移	(148)
8.3.2 旋转	(149)
8.3.3 缩放	(150)
8.3.4 操作矩阵	(152)
8.4 填充风格	(154)
8.4.1 渐变填充	(154)
8.4.2 图案填充	(156)
8.5 阴影效果	(157)
8.6 合成	(159)
8.7 剪切	(160)
8.8 状态方法	(161)
8.9 小结	(163)
8.10 习题	(163)
第 9 章 CVIDrawJS 绘图部分	(164)
9.1 图形对象 CVIGraph	(164)
9.1.1 嵌入引擎脚本	(165)
9.1.2 创建图形对象的实例	(166)
9.1.3 几何对象和颜色对象	(166)
9.2 绘图属性	(167)
9.2.1 线条属性	(167)
9.2.2 填充属性	(168)
9.2.3 全局属性	(170)
9.3 绘图方法	(171)
9.4 小结	(176)
9.5 习题	(176)

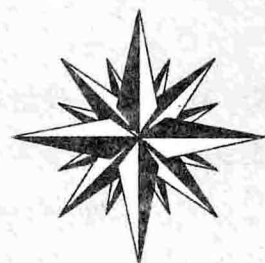


第三部分

第 10 章 预备知识	(179)
10.1 动画概述	(179)
10.1.1 setInterval()函数	(180)
10.1.2 平移动画	(181)
10.1.3 精灵动画	(182)
10.2 提高绘图效能	(184)
10.2.1 预渲染	(184)
10.2.2 多层画布	(186)
10.3 消息响应	(187)
10.3.1 鼠标响应	(187)
10.3.2 鼠标事件中的属性	(189)
10.3.3 简单画板	(190)
10.3.4 键盘响应	(192)
10.3.5 键盘事件中的属性	(193)
10.4 设备事件	(197)
10.4.1 触摸与手势事件	(198)
10.4.2 方向事件	(200)
10.5 小结	(200)
10.6 习题	(201)
第 11 章 HTML5 动画设计	(202)
11.1 鱼游动动画设计	(202)
11.1.1 精灵作平移动画	(202)
11.1.2 添加上下移动动作	(204)
11.1.3 面向对象编程实现	(205)
11.1.4 修改 Fish 对象	(208)
11.1.5 添加文字	(211)
11.2 广告动画	(214)
11.3 参数约束的动画实例——曲柄滑块结构	(217)
11.3.1 什么是曲柄滑块结构	(217)
11.3.2 曲柄滑块结构动画设计思路	(218)

11.3.3 静态帧的绘制	(220)
11.3.4 完整动画	(221)
11.4 小结	(223)
11.5 习题	(224)
第 12 章 HTML5 休闲游戏设计	(225)
12.1 游戏策划	(225)
12.2 碰撞检测	(226)
12.2.1 碰撞检测类型	(226)
12.2.2 碰撞检测与鼠标交互	(227)
12.2.3 实现拖动效果	(229)
12.3 捕鱼小游戏设计	(233)
12.3.1 添加拖动效果	(233)
12.3.2 效果调整	(238)
12.3.3 收获鱼儿	(242)
12.3.4 完善游戏	(245)
12.4 小结	(249)
12.5 习题	(249)
参考文献	(250)

第1章 基本概念



第一部分

第1章 基本概念

随着时代发展,移动性和跨平台性成为当前的一大发展趋势,围绕着这一发展趋势,产生了许多新的技术。其中,基于 Web 浏览器的 HTML5 技术受到了广泛关注。下面简单介绍 HTML5 的特性以及用于浏览器客户端的脚本语言 JavaScript。

1.1 了解 HTML5

1.1.1 什么是 HTML5

HTML 是 Hypertext Markup Language 的缩写,即超文本标记语言。超文本标记语言是标准通用标记语言下的一个应用,它通过标记符号来标记要显示的网页中的各个部分,告诉浏览器如何显示其中的内容。

HTML 标准自发布以后,相当一段时间都没有推出新的标准。各家浏览器厂商为了满足日益增长的网站应用功能,联合部分公司和组织创建新的标准,此即为 HTML5 的前身。提出草案并获得万维网联盟(W3C)接纳后,命名为 HTML5 新标准并成立了工作团队。目前,HTML5 标准还在不断发展并完善着。

目前支持 HTML5 的浏览器包括 Firefox、IE9 及其更高版本,Chrome、Safari、Opera 等,而国内的傲游浏览器,以及基于 IE 或 Chromium 所推出的 360 浏览器、搜狗浏览器、QQ 浏览器、猎豹浏览器等浏览器同样具备支持 HTML5 的能力。

1.1.2 HTML5 新特性

HTML5 在原先 DOM 的基础上,增加了多样化的 API 函数,这些新特性让 Web 前端开发人员能够更加轻易地添加很多炫酷的网页效果。新特性包括:

- ❖ 用于绘画的 canvas 元素



- ❖ 用于媒介回放的 video 和 audio 元素
- ❖ 对本地离线存储的更好的支持
- ❖ 新的特殊内容元素
- ❖ 新的表单控件

其中最让移动端广告动画及游戏开发者感兴趣的就是 HTML5 中新增的 3 个标签, 分别是 `<canvas>`、`<video>` 和 `<audio>` 标签。虽然支持 HTML5 的浏览器都支持这三种标签, 但是由于图片、音频和视频的格式众多, 并不是每个浏览器都能支持多种的多媒体格式。

1.1.3 HTML5 发展趋势

1. 移动优先

从如今层出不穷的移动应用就知道, 在这个智能手机和平板电脑大爆炸的时代, 移动优先已成发展趋势, 不管开发应用还是游戏, 都必须考虑移动设备的跨平台性和兼容性。

2. 适合游戏开发

由于 HTML5 的跨平台性, 一次开发游戏, 可适应不同的平台, 目前游戏开发商都愿意用 HTML5 来开发游戏。通过 PhoneGap 及 appmobi 的 XDK 将 Web 游戏应用打包整合到原生应用中也是一种应用方式。

3. 2014 计划

2012 年 9 月, 万维网联盟 (W3C) 提出计划要在 2014 年年底前发布一个 HTML5 推荐标准, 并在 2016 年年底前发布 HTML5.1 推荐标准。

1.2 了解 JavaScript

1.2.1 什么是 JavaScript

JavaScript 是一种基于对象和事件驱动的客户脚本语言。同时也是一种广泛用于客户端 Web 开发的脚本语言, 常用来给 HTML 网页添加动态效果和处理交互逻辑, 比如修改文档内容和响应用户的交互操作。

完整的 JavaScript 实现包含三个部分:

- ❖ 核心(ECMAScript)
- ❖ 文档对象模型(DOM)
- ❖ 浏览器对象模型(BOM)

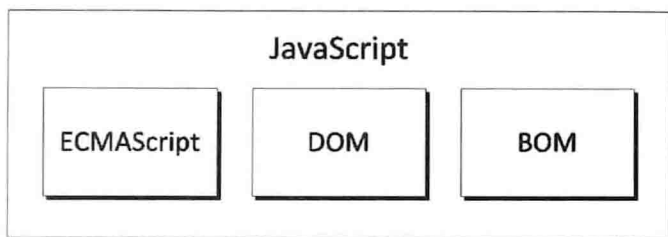


图 1-1 JavaScript 组成

1.2.2 核心(ECMAScript)

ECMAScript 与 Web 浏览器并没有直接的依赖关系。ECMAScript 主要向 JavaScript 脚本语言提供了核心类型和基本语法,以下是 ECMAScript 规定的内容。

- ❖ 语法
- ❖ 类型
- ❖ 语句
- ❖ 关键字、保留字
- ❖ 操作符
- ❖ 对象

学习 JavaScript 的过程其实就是先学习 ECMAScript 提供的基本语法和语句,因此有时 JavaScript 和 ECMAScript 被误以为是相同的含义。

1.2.3 文档对象模型(DOM)

文档对象模型(DOM)是一种用于 HTML 和 XML 文档的编程接口。DOM 为一个多层节点结构。HTML 或 XML 页面中的每个组成部分都是某种类型的节点,而这些节点又包含着不同类型的数据。通过 DOM 可以动态地访问程序和脚本,更新其内容、结构和 WWW 文档的风格。文档可以进一步被处理,处理的结果可以加入到当前的页面。

DOM 是一种基于树的 API 文档,它要求在处理过程中整个文档都表示在存储器中,如图 1-2 所示。