

网页游戏 开发秘笈

(美) Evan Burchard 著 爱飞翔 译

The Web Game Developer's Cookbook

Using JavaScript and HTML5 to Develop Games

- 资深网页游戏开发专家亲笔撰写，众多游戏开发与设计专家强力推荐，Amazon全五星评价
- 精选10种常见的游戏类型，透过典型实例深入剖析游戏引擎及工具的选用技巧，详细讲解每款游戏的制作过程，为快速掌握网页游戏开发提供系统实用指南



机械工业出版社
China Machine Press

网页游戏 开发秘笈

(美) Evan Burchard 著 爱飞翔 译

.....
The Web Game Developer's Cookbook
Using JavaScript and HTML5 to Develop Games
.....



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

网页游戏开发秘笈 / (美) 伯查德 (Burchard, E.) 著; 爱飞翔译. —北京: 机械工业出版社, 2014.4

(游戏开发与设计技术丛书)

书名原文: The Web Game Developer's Cookbook: Using JavaScript and HTML5 to Develop Games

ISBN 978-7-111-45992-7

I. 网… II. ①伯… ②爱… III. 游戏程序—程序设计 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2014) 第 037334 号

本书版权登记号: 图字: 01-2013-4457

Authorized translation from the English language edition, entitled *The Web Game Developer's Cookbook: Using JavaScript and HTML5 to Develop Games*, 9780321898388 by Evan Burchard, published by Pearson Education, Inc., Copyright © 2013.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese Simplified language edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2014.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括中国台湾地区和香港、澳门特别行政区) 独家出版发行。未经许可者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 防伪标签, 无标签者不得销售。



网页游戏开发秘笈

(美) Evan Burchard 著

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 关 敏

印 刷: 蕺城市京瑞印刷有限公司

版 次: 2014 年 5 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 19

书 号: ISBN 978-7-111-45992-7

定 价: 69.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

译者序

最近很多人开始关注网页游戏了。从游戏开发的角度来看，网页游戏这一形式具有诸多优点。

首先，传统的游戏开发形式大多受制于移植问题。开发好一款游戏之后，需要将其移植到多个操作系统中，而每个操作系统所适宜的开发环境又各有区别。网页游戏则不然：开发者把大部分精力集中在浏览器里即可。

此外，传统的开发形式一般需要大量的资金和人员支持，而网页游戏则特别适合中小团队及独立游戏开发者。从创意，到实现，再到测试并发布，这个周期可以缩得很短，而且过程也可以很灵活，发现新想法之后，立刻就能实验并看出效果来。

但是，初学者在入门时会遇到几个困难，其一是 JavaScript 语言与 HTML5 的特性太多，短期内很难将这些知识点全部掌握，而且有些特性与游戏开发的关系并不是很大。其二在于，许多读者原来未必具备丰富的开发经验，大家可能是从其他编程语言、其他开发平台，甚至其他行业转入网页游戏开发领域的。我们必须找到一套实用的开发流程，否则每次做游戏都要从头写起，这会耽误大量时间。

本书就相当顺畅地解决了这几个问题。作者 Evan Burchard 先生没有讲述高深的理论，而是直接选了 10 种常见的游戏类型，告诉我们如何通过适当的游戏引擎及工具，快速制作出这些游戏来。每章所选的范例游戏，其制作过程都分为好几个步骤，读者可在看完每个步骤之后及时总结当前制作进度。这些范例其实就是模板，只要根据每章最后的建议稍加修改，就能做出一款颇具个人特色的网页游戏了。

通过引擎来做游戏，既能缩短学习时间，又能降低编写代码的难度，而且只要学会一种引擎，就可以开发出许多款同类游戏了。由此可以想见，在学完全部 10 款引擎之后，你的开发思路一定会大为开阔。

这本既直观又实用的教程，不仅对初学者有用，中等水平的读者也能从中收益。你可以对比书中所用的引擎与你所喜好的引擎之间有何异同，也可以思考怎样用引擎来制作书中没有讲到的那些游戏类型，还可以研究书中所提到的各种游戏算法及游戏创意。

要提醒大家注意的是，与具有深厚积淀的传统技术领域不同，网页游戏是个变化很快的行业，各种 HTML5 新特性层出不穷，而且 JavaScript 语言标准、程序库、游戏引擎等也都日新月异。一方面要熟悉 JavaScript 语言、努力提高代码质量，另一方面也要紧跟潮流，多看、多学、多练，不断提升开发熟练度及创意能力，有条件的朋友还可适当参与开源项目。针对这些内容，作者在书后总结了三个很有参考价值的附录，可供大家随时查阅。

本书翻译过程中，得到了机械工业出版社华章公司诸位编辑与工作人员的帮助，在此深表感谢。

由于时间仓促，译者水平有限，错误与疏漏之处在所难免，敬请读者批评指正。大家可发邮件至 eastarstormlee@gmail.com 与我联系，也可访问网页 <http://agilemobidev.com/eastarlee/book/the-web-game-developers-cookbook> 留言。

爱飞翔

2014 年 3 月

前言

笔者小时候的一件乐事就是玩日本出的游戏卡带（cartridge），那上面都有“任天堂公司官方品质认证标贴”（Official Nintendo Seal of Quality），而且带着一股塑料味儿。把这些游戏卡插在一个“魔盒”^①里，并按下“电源”键，然后，充满挑战、发现与征服的娱乐之旅就开始了。后来我发现了一件令自己颇感吃惊的事：其实我玩的这些游戏（以及其他类似游戏）普通人也可以做出来，而且有的只需一个人或几个人就够了，但这些制作团队却变得越来越大了。正如笔者所见，当初那些由几个游戏迷组建的小团队的行业，现今正成长为价值 500 亿美元的电子游戏业。

时下，虽然大型游戏工作室已经占据市场主导地位，但是小型的独立游戏开发团队依然有重新焕发生机的机会。这些团队在发布游戏时有许多平台可供选择，然而那些平台中所发生的革新，都不如 Web 浏览器领域这般显著，这个平台原本比较低调，而且容易为人忽视。但现在，随着浏览器技术的发展，涌现出数以百计的免费游戏引擎，通过这些引擎，游戏设计者只需独自一人，即能创建出具备个人风格的游戏来，这些游戏可以做得非常有趣，给玩家留下深刻印象，甚至还有潜在商机。要制作这种网页游戏，只需浏览器和文本编辑器，并掌握本书所讲的一些知识即可。偶尔需要打开控制台，不过更多的时候只用点几下按钮就行，想制作一款能给自己和他人带来乐趣的游戏，这是最简单不过的办法了。

快按下电源键，开始跟我学做游戏吧！

致谢

首先，真诚地感谢每位读者。你肯阅读本书，笔者特别开心。非常感谢。

还要感谢 Pearson 公司的工作团队，尤其是 Laura、Olivia 和 Songlin 三位，感谢你们给我机会写作本书，并指导我写完。

感谢诸位友人及审阅者：Jon、Rich、Jason、Greg、BBsan、Pascal、Tim 和 Tony。

① 指 FC 游戏机之类的家用游戏主机。——译者注

感谢聪慧的母亲，感谢洞察秋毫的父亲，感谢耐心而卓识的 Amy。感谢 Gretchen 与 Max，谢谢你们作为首轮测试者，来玩我所开发的游戏，你们真是相当率直而喜乐之人。

在成长过程中，有很多好游戏伴随着我，所以要感谢这些游戏的每位制作者。20 世纪 90 年代，有很多 ROM 破解社区，我从这里初次了解到如何剖析游戏，所以要感谢其中的每位成员。

感谢开源社区的贡献者。你们为这个世界贡献了许多精彩代码，正因为受你们影响，我才会加入开源者的行列，并享受个中乐趣。笔者在本书中用到了一些工具（参见附录 C），尤其要感谢这些工具的作者。若没有这些工具，本书绝对无法完成。在 HTML5 游戏制作的合成与展示方面，Kesiev 做了许多工作，我对此表示特别感谢。

感谢 Morris 先生边看我写书，边给我挑毛病；感谢 Jamison 博士教我领会了“理解的广度与深度”是何等重要；感谢 Hatasa 博士给我提供了一个全新的视角，令我重新审视这个世界。

感谢所有唱诗班与剧场里的孩子们，也感谢里面的诸位朋克、怪咖、书呆子、极客、工程师、企业家、研究者、设计者、梦想家和百事通，这些年来，是你们令我保持良好心态，你们给我带来了欢乐，也带来了恰到好处的小小烦恼。尤其要感谢剧场里那一位能耐心忍受我长时间纠缠的小朋友。

最后，感谢信任我并给我理由的每个人，也要感谢虽不信任我却能给我理由的每个人。

导 言

从前制作游戏时经常需要使用很多特殊工具。现在不同了，只需浏览器和文本编辑器就行。不仅仅是 HTML5 游戏，采用其他技术来制作游戏所需的时间与成本也比原来大大降低了，现在只需几天，甚至几小时就能做好一款游戏。独立游戏开发者的参与平台正在渐渐扩大，比如“游戏制作节”(game jam)，这是一种以多人在线协作方式迅速制作游戏的开发者集会。

game jam 通常为 48 小时，大型的在线开发者聚会活动（如 Global Game Jam 与 Ludum Dare 等）都是这么长的时间。不过游戏设计师们都喜欢自创规则，所以有些 jam 可以短至仅 1 小时。除了能增进沟通及协作能力之外，游戏制作者还能迫使自己练就迅速制作游戏的本领，以便下次能更快做好游戏，这项本领对长期与短期游戏项目都有好处。

不是只有独立游戏开发者才需要“快速制作东西”这项技能。软件公司也需要，他们把这称为“生产力”。有句口号叫做“变得更聪明”，可惜这句话说得很不明确，其实，要想提高工作效率，“找对工具并善用工具”才是更高明的办法。这句话所传达的信息更具说服力，要是把数学等知识也包括在“工具”一词的定义中，那就更加能令人信服了。

笔者起初列了份清单，其中有一百多个 HTML5 游戏引擎，成书时将其缩减，只保留了最优秀的几个。通过这些引擎，以及本书中提供的工具，就能用浏览器快速制作出游戏来了。使用这些引擎很简单，只需把对应的 JavaScript 代码载入 HTML 文件即可，偶尔可能需要添加一两行代码。总的来说，笔者选定的这些引擎都有良好的开发文档和活跃的支持社区。某些引擎可能比其他引擎要大一些。它们都提供了开发游戏所需要的特有功能，等学过几个引擎后，你就能体会出各引擎之间的异同了。

本书每章讲解一个引擎，并选一种游戏类型与之相配。读者在阅读过程中会发现，随着游戏类型越来越复杂，所选的游戏引擎也必须功能越来越丰富才行。读完本书之后，再学其他游戏引擎就非常轻松了，你甚至可以自己拼装一个引擎玩玩。

本书所讲的每个游戏都只需几个小时就能做好。那大家可能要问了：每个类型里所举的那个范例游戏，会是我所喜欢的吗？笔者觉得……很有可能不是。本书要做的是剖析这些游戏类型，将每

个类型都分解成几个基本部件。这就好比盖房子，本书只是打好地基，立好框架，竖好白墙。某些情况下笔者可能会略加点缀。然而尚有不尽如人意之处，比如屋顶可能会有个大洞，比如墙上可能挂的是笔者喜欢的画作而非你自己喜欢的那幅。没关系，你可以自己动手，比方说，设个庭院，铺上粗毛线毯，再种几株银杏。可以把我挂的那幅画换成你自己喜欢的。我只是告诉你从哪里获取相关素材而已，至于具体如何布置，那都由你决定。本书就是如此：读者按照自己的需要来行事即可，只要把自己想要的东西添上去，这就是你自己的游戏啦。

读完本书之后，大家很容易就能构想出一款自己所中意的游戏，将其拆解为若干功能，并使用本书所讲的这套工具，以相似的流程把它做好。大家甚至都能预估出游戏的制作难度和所需时长。如能善用工具，并且创意颇佳，那我想你应该能制作出一款好游戏，把大家立刻迷住喔。

读者对象

若你打算阅读本书，我想可能有几个原因。如果你是对游戏感兴趣，并想学着编程，那这本书很合适。如果你是一名网页开发者或设计师，正在寻找制作游戏所需的工具、技术与模板，或者你是一位 **JavaScript** 程序员，想从入门级提升为中级水准，那么这本书也很合适。如果你是一名游戏设计者或开发者，原本在制作 **flash** 游戏、移动 / 桌面端的原生游戏，或其他平台的游戏，而现在想来看看怎样在 **HTML5/JavaScript** 环境下开发游戏，那么这本书还是很合适。但如果你把 **HTML5** 盾形图标纹在身上，经常展示自己为开源社区贡献的游戏引擎代码，而且只花一个周末就能用 **HTML5** 开发出“**Mario 64**”这种游戏，并能将其移植到 **iPhone** 上面，那恐怕就不需要读这本书了吧。

本书代码风格及行文约定

新加入的或有所更改的代码行，用粗体标出。程序清单里省去的一行或多行代码，以省略号 (...) 标识。如需刻意指明某行已经删掉或改掉的代码，则以注释形式（以 // 开头）将原来的代码写在删改前的位置上。若整段程序清单都是新代码，则不加粗。

接续符 (➡) 表明当前这行代码是接着上一行代码来写的。

提示

某些需要稍加解释的内容，会写在这种“提示”框里面。

技巧

那些值得一提但却不适合放入正文的内容，作为“技巧”列于此处。

警告

某些不太明显的地方，或是在不知情时容易出问题的地方，写在“警告”框里面。

本书内容组织方式

本书分为 11 章，从第 1 章~第 10 章每章都讲一个游戏，最后有三个附录（附录 A 讲解了 JavaScript 的基础知识，附录 B 讲解了管控代码质量的方式，附录 C 列出了制作游戏时要用到的资源）。读者在阅读第 1 章时，不需要任何 HTML、CSS 或 JavaScript 基础，也不要求会使用各种工具。然而在阅读后续各章之前，最好能先把附录 A 与第 1 章掌握了。从代码角度来讲，后续每一章都不需要用到前面各章的代码。不过，各章所讲述的游戏类型渐趋复杂，所以最好是先把比较简单的几种游戏类型学会，然后再学后面的章节，这样会更顺利些。第 11 章可以作为一份指导材料，它告诉大家读完本书之后应该继续学习哪些内容。最后的附录 C 里还有一份资源列表可作为补充材料，其中列出了本书第 1 章~第 10 章创建游戏时所用的工具。

每个游戏的制作过程都分解为数个步骤，而每一步又细分为若干段代码与文本，这样便于大家理解。这些步骤所对应的源文件均可在 jsarcade.com 网站中找到。也就是说，每一步所对应的源文件都放在一个文件夹中，而这些文件夹及其中的代码均可从本书英文版配套网站里下载。如果在阅读过程中“迷了路”，或是想跳读，那么可以抛开当前这一步的代码，直接从后续步骤的代码开始研究。有时也许想看看这个游戏的最终成品是什么样，那就可以直接调到“final”目录，这里列出了每章所做范例游戏的最终版本。

在掌握了第 1 章与附录 A 之后，如果对本书其他各章还是看不太懂的话，那么可以参考附录 B，其中讲了很多避免编程困境的办法，以及遭遇编程难题时的解决方式。

本书阅读方式

为了充分理解书中内容，需要下载每章的源代码。这些源码包括 JavaScript、HTML、CSS、图片，以及每一步所需的全部附加文件。这些都能在 jsarcade.com 上找到。代码首先按每章章名分成数个文件夹。每章所对应的文件夹里，都包含运行本章游戏所需的全部代码，这些代码会放在三种不同类型的目录中。“initial/”目录里含有运行游戏所需的最基本代码。“after_recipe<x>/”目录中包含执行完每一步之后的“成果”[⊖]（基本上对应于每一章内的节标题），这样的话，即便在阅读时乱了思路，也没有关系，因为顶多只有一两页稍微看不明白而已，你马上就能在下个步骤开始时重新整理思路。“final/”文件夹下的代码表示每章范例游戏的最终状态。在每一章中，每个步骤所对应的文件夹里都有名为 `index.html` 的文件。可以双击打开该文件，也可以通过其他方法将其放在浏览器中运行，此时看到的效果就是执行完该步骤后游戏所应有的样子。每个范例游戏的最终版本均可以在 jsarcade.com 网站上找到，可以访问该网站，预览一下每个游戏，然后从中选定自己喜欢的游戏类型并实现它。

⊖ 原文为“checkpoint”，指检查点、关键点、基准点。——译者注

**提示**

所有游戏源码、游戏引擎及其他相关软件均可在 jsarcade.com 网站或 [informit](http://informit.com) 网站 (informit.com/title/9780321898388) 中下载。

本书可以跳读，然而要知道，每一章所讲的游戏类型都会比上一章复杂。如果碰到无法理解的地方，可以查看每个步骤所对应的成果代码（也就是 `after_recipe<x>` 文件夹中的代码），而且还需特别留意第 1 章与附录 A，看看自己是否已经掌握了其中的内容。如果在制作游戏的过程中出了错，而又不明白其原因，那么请阅读附录 B。

学完一章之后，你也许会觉得游戏中好像还是缺了点儿什么似的。你可能还想实现一个爆炸效果，还想添加一出精彩的剧情，还想设计一场“Boss 战”(boss battle)[⊖]。每章结尾都给出了诸如此类的建议，你可按照这些建议来改进游戏，如果你有不同的见解，也可以按自己的想法行事。只要在电脑中运行了这些游戏代码，它就算是你自己的作品了。这些代码本来就是模板，本来就是可以修改、扩充，并按自己喜好来定制的。若你修改之后的游戏比笔者原先写的还要好，那我绝不会嫉妒，而只会乐见其成的。

⊖ “boss” 可理解为游戏中的妖怪头目或敌方首领。——译者注

目 录

译者序

前 言

导 言

第 1 章 问答游戏	1
1.1 第一步：出题	1
1.2 第二步：隐藏题目	8
1.3 第三步：重现题目	9
1.4 第四步：引入各种程序库	11
1.5 第五步：判断玩家所选答案是否正确	15
1.6 小结	19
第 2 章 文字冒险游戏	20
2.1 第一步：设计页面样式	21
2.2 第二步：实现页面跳转	25
2.3 第三步：添加道具栏及道具拖放功能	27
2.4 第四步：添加复杂的交互功能	35
2.5 第五步：添加历史记录导航功能	44
2.6 第六步：添加精彩的结局	46
2.7 小结	48
第 3 章 派对游戏	50
3.1 第一步：采用 atom.js 创建范例游戏	50

3.2	第二步：用 <code>canvas</code> 元素绘图	56
3.3	第三步：绘制鼠洞	58
3.4	第四步：绘制鼯鼠	60
3.5	第五步：将鼯鼠放入鼠洞	63
3.6	第六步：令鼯鼠从洞中钻出来	65
3.7	第七步：使玩家可通过敲击键盘来打鼯鼠	67
3.8	HTML5 的 <code><audio></code> 标签并不尽如人意	70
3.9	小结	72
第 4 章 解谜游戏		73
4.1	第一步：用 <code>casel.js</code> 来渲染	74
4.2	第二步：渲染多个方块	78
4.3	第三步：创建成对出现的同色方块	81
4.4	第四步：配对并消除同色方块	83
4.5	第五步：隐藏与翻转图片	85
4.6	第六步：胜负判定	87
4.7	第七步：缓存与性能优化	91
4.8	第八步：将方块配对游戏改为字母认读游戏	93
4.9	小结	99
第 5 章 平台游戏		100
5.1	初识 <code>melon.js</code>	100
5.2	第一步：创建瓦片地图	101
5.3	第二步：启动游戏	102
5.4	第三步：加入游戏角色	105
5.5	第四步：构建碰撞图层	107
5.6	第五步：行走与跳跃	108
5.7	第六步：标题画面	110
5.8	第七步：添加可供玩家收集的物品	112
5.9	第八步：添加敌人	113
5.10	第九步：添加能力提升道具	115
5.11	第十步：添加胜负判定机制并显示游戏信息	116

5.12 小结	118
第 6 章 格斗游戏	120
6.1 第一步：初识 <code>game.js</code>	120
6.2 第二步：单独绘制精灵表中的各个精灵	123
6.3 第三步：处理两位玩家的输入	124
6.4 第四步：移动与变形	127
6.5 第五步：不要使两位玩家的输入互相阻塞	130
6.6 第六步：用位掩码记录按键状态	134
6.7 第七步：用掩码来检测碰撞	137
6.8 第八步：令两个游戏角色可以互相攻击对方	140
6.9 小结	145
第 7 章 飞行射击游戏	146
7.1 与渲染有关的准备知识	146
7.2 第一步：初识 <code>gameQuery</code>	148
7.3 第二步：添加“敌机”	150
7.4 第三步：创建己方战机	153
7.5 第四步：处理玩家与敌人的碰撞	156
7.6 第五步：发射子弹	157
7.7 第六步：为玩家提供多种武器	159
7.8 小结	162
第 8 章 第一人称射击游戏	164
8.1 第一步：初识 <code>Jaws</code>	164
8.2 第二步：创建二维地图	166
8.3 第三步：增添玩家	169
8.4 第四步：用光线投射技术绘制俯视图	173
8.5 第五步：用光线投射技术实现伪三维效果	177
8.6 第六步：为玩家配备照相机	180
8.7 第七步：装点游戏场景	184
8.8 第八步：添加同伴	188
8.9 小结	195

第 9 章 角色扮演游戏	197
9.1 第一步：初识 <code>enchant.js</code>	197
9.2 第二步：创建地图	199
9.3 第三步：增添玩家	202
9.4 第四步：添加碰撞层	207
9.5 第五步：绘制状态画面	209
9.6 第六步：与 NPC 对话	212
9.7 第七步：绘制道具栏	215
9.8 第八步：开设道具店	218
9.9 第九步：实现战斗界面	226
9.10 第十步：通过 <code>HTML5</code> 的本地存储 API 来存储游戏进度	236
9.11 小结	238
第 10 章 即时战略游戏	240
10.1 服务器	240
10.2 第一步：配置 <code>Node</code>	243
10.3 第二步：用 <code>socket.io</code> 实现即时通信	246
10.4 第三步：用 <code>crafty.js</code> 创建 <code>Isometric</code> 地图	249
10.5 第四步：绘制棋子	251
10.6 第五步：移动棋子	255
10.7 第六步：使玩家只能控制己方棋子，并将对方棋子上面的图案隐藏起来	258
10.8 第七步：实现碰撞检测，令玩家能吃掉对方棋子并看到其图案	264
10.9 小结	268
第 11 章 如何继续提升游戏开发水平	270
11.1 本书已经讲过的内容	270
11.2 接下来应该学习的内容	271
附录 A JavaScript 基础	273
附录 B 代码质量控制	280
附录 C 资源	285

第 1 章

问答游戏

此类游戏规则很简单：在数个选项中找出问题的正确答案。如果不知道的话，随便猜一个也行。从酒馆中的竞猜到 SAT[Ⓐ]，到处都能见到这种游戏的身影，说它是“无孔不入的”游戏类型，其实都低估了其流行程度。电视娱乐节目里也有互动形式的问答游戏。有时除了简单的问题与答案之外，游戏里可能还有其他元素，不过这些游戏的底层逻辑都一样。比方说，在某个游戏里，国王询问玩家想不想与恶龙决斗，玩家回答“想”，那这就可以算作一个相当短小而简单的问答游戏（quiz）。说得更宽泛一些，在 RPG（Role-Playing Game，角色扮演游戏）中，玩家从平台掉入洞穴里，或者失掉了全部生命值（hit point），也可以看成是对答错问题的一种惩罚。不管问答机制出现在哪一类游戏里，我们都得用相似的代码来实现这套规则，并设计好答对或答错问题之后所要执行的逻辑。

1.1 第一步：出题

由于大家的网页开发水平各不相同，所以笔者把第 1 章尽可能写得简单明了，以便入门级的读者能够看懂。后续各章会越来越复杂，不过理解了本章之后，再去看那些章节，也就不会太难了。每个人对网页开发的掌握程度不一样，有些读者或许真的要从这里开始学起。如果觉得本章内容过于简单，那就可以略读或直接跳过。后面各章会比本章复杂，也会更难一些。

本章通过制作范例游戏，想达成三个主要目标。第一，要使大家理解 HTML、CSS 和 JavaScript 的基础知识。在这三者之中，JavaScript 最难。若是对某些 JavaScript 基础知识还不甚了解的话，请参阅附录 A。第二，要令读者学会如何引入程序库，因为本书将要用到很多外部程序库。第三，要使大家掌握一套便捷的开发流程，用来创建、编辑、保存和打开本

Ⓐ SAT 曾经是“Scholastic Aptitude Test”（学术能力测验）或“Scholastic Assessment Test”（学术评估测试）的简称，这种测验是美国各大学决定是否录取申请者的重要参考指标之一。——译者注

书所用的各种文件，并反复运用此流程，以制作各类游戏。

如果没有文本编辑器，那现在就得找一个。本书用到的这些 JavaScript、HTML 和 CSS 文件，拿任何工具来创建或编辑都行。大家若还没找到合适的文本编辑器，那就请参考附录 C，根据其内容选择一款。

现在开始做游戏。首先打开文本编辑器，将程序清单 1.1 中的代码加入 quiz/initial/index.html 文件中。如果还没下载这份代码文件，那么请先参阅导言部分，按照其中所给的网址将其下载下来。

程序清单 1.1 在 index.html 文件中建立 html 结构

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Quiz</title>
    <link rel="stylesheet" type="text/css" href="main.css">
  </head>
  <body>
    <h1>Quiz</h1>
    <div id="quiz">
      </div>
    </body>
  </html>
```

提示

HTML 是 HyperText Markup Language（超文本标记语言）的缩写。从前，“链接”（link）曾经叫做“超链接”（hyperlink），那时还有几个与“hyper”有关的东西，用于在文档之间跳转。“超文本”（HyperText）可以理解为含有超链接的普通文本。而“标记”（markup）则是包裹在超文本周围的辅助文本，用来赋予其更多含义。所以说，HTML 是一系列语法指南，用于将不同类型的文本制作成可以互相链接的页面，并将其组合起来，而这些页面文件以 .html 为扩展名。

<以尖括号形式出现的代码>叫做 HTML 标签（tag）^①，而在<开始标签>（beginning tag）与<结束标签>（closing tag）^②之间的代码则称为 HTML 元素（element），元素也包括首尾两个标签在内。注意，结束标签里要写上“/”字符。

首先声明 DOCTYPE。浏览器看到此声明之后，就知道应该按照 HTML 文件来解析并呈现其内容了。由于浏览器还可以打开其他类型的文件，比如 XML 文档、音频文件和图片等，所以，为了使浏览器能按普通的网页来处理此文件，我们需要为它加上这条声明。如果不加这一行，会怎么样呢？这取决于你所使用的浏览器，在某些浏览器上可能会产生难于察

① 为了与“markup”（标记）相区分，这里将“tag”一词译为“标签”。——译者注

② 也称“起始标签”与“闭合标签”。——译者注