

产品 + 交互 + 前端

全方位提升产品多平台适应性

打造历久弥新的超凡用户体验

渐进增强

跨平台用户体验设计

[美] Aaron Gustafson 著 翟东方 任洁 译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

渐进增强

跨平台用户体验设计

[美] Aaron Gustafson 著 翟东方 任洁 译



人民邮电出版社
北京

图书在版编目 (C I P) 数据

渐进增强：跨平台用户体验设计 / (美) 亚伦·古斯塔夫森 (Aaron Gustafson) 著；翟东方，任洁译. --
北京：人民邮电出版社，2017.3
ISBN 978-7-115-44848-4

I. ①渐… II. ①亚… ②翟… ③任… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆CIP数据核字(2017)第024646号

版 权 声 明

Authorized translation from the English language edition, entitled ADAPTIVE WEB DESIGN, 2nd Edition, 9780134216140 by GUSTAFSON, AARON, published by Pearson Education, Inc, publishing as New Riders, Copyright © 2016.

本书中文简体字版由美国 Pearson Education 授权人民邮电出版社出版。未经出版者书面许可，对本书任何部分不得以任何方式复制或抄袭。版权所有，侵权必究。

-
- ◆ 著 [美] Aaron Gustafson
 - 译 翟东方 任洁
 - 责任编辑 赵轩
 - 责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京画中画印刷有限公司印刷
 - ◆ 开本：720×960 1/16
 - 印张：12.75
 - 字数：215千字 2017年3月第1版
 - 印数：1—3000册 2017年3月北京第1次印刷

著作权合同登记号 图字：01-2016-2418号

定价：59.00元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315

内容提要

大多数有关 Web 设计的图书都采取了“技术 + 案例”的形式，以便让读者迅速掌握某项技能。这类图书会用很大篇幅讲述如何应用 HTML 标签，CSS 代码的每个属性是如何生效的，等等。此外，这些图书还会附带一个一个的小项目，通过带着读者一步一步写出代码，来观察项目是如何成型的，让读者形成自己的认识，从而掌握设计和开发技巧。

这本书和上面所说的模式不同。这本书讨论的是设计师如何应对一个不断发展变化的 Web 环境，本质上，这是一本设计哲学书。

市面上已经存在大量的技术图书，网上也有大量的技术文章可以读。这些书很多写得都非常好，但是，如你所知，大多数图书和文章在几个月到一年的时间之后就被淘汰了，因为技术过时了。

实现技术、浏览器、工具包……这些东西变化得太快。我自己经常因为学习这些东西导致不堪重负。在一片动荡不安的技术海洋里，到底有多少种方法可以学习啊！我只是想做一个小网站，这片大海简直把我吓坏了！当我被技术的浪潮推到这边，拍到那边，因为晕船吐得一塌糊涂之后，我终于发现了大多数技术都在围绕一个中心在发展自己——这个中心就是“渐进增强”。

“渐进增强”的哲学可以称之为适应性 Web 设计方法的核心和灵魂。以它为核心，我终于能够正确理解很多技术和思路是如何产生的。甚至，理解了这种思想之后，我做出来的新网站变得更加健壮，而且很多用户表示我的网站访问起来终于不那么头疼啦，因此，这些网站的访问量也一天天增长起来。对“渐进增强”的理解使我成为了一名更优秀的 Web 设计师，我相信这种魔力也会让你变得更优秀。

序

至今我依然清楚地记得，本书第一版刚刚面世，我用双手捧着它的那种感觉。我早知道这是本好书，彼时的我就像孩子剥开糖果一样，期待着满心甜蜜。结果，这本书的第1章就完全超过我的预期，让我佩服得五体投地。在第1章中，Aaron明确而具体地解释了那个我为之思考多年，却总是不能想明白说清楚的设计原则——渐进增强的真谛。

事后想来，我本不该如此惊讶。Aaron本是一个Web领域的多面手，在很多方面都有深入的研究——特别是可用性方面。但是他并没有像很多人一样，做一个技术领域的专家，而是不吝把他的思想分享给更多的人，让更多的人受益。

对于我们这些Web设计师和开发者来说，跟随技术趋势、学习实现技巧就像家常便饭一样，我们也乐得沉溺其中，被技术左右，为一些效果的实现方式而百思不解。我们常常忘记问自己一个根本问题，Web的设计和开发最初是为了什么呢？而Aaron除了能够引导读者进入技术的细微末节之外，关键他能够时刻提醒你，时刻不要忘了，技术只是产品的一部分，你的产品设计最终还是要考虑到用户的因素。

我时常怀疑，Aaron是不是会纠结于如何描述自己的工作性质。开发者？布道者？作者？所有这些名词都只描述了他工作的一部分，我们无法通过这些头衔来真正了解他做的事情。我觉得，Aaron最适合的头衔也许是——老师。

一位好老师总是具有神奇的魔力。他能够一边传授知识一边像讲故事一样娓娓道来，让听众欣然接受。Aaron的这本书正是具有这样的魔力。

我相信你会喜欢这本书的。像读故事书一样翻开后面的篇章吧，你收获的可不仅仅是故事。

开始吧，老师！

Jeremy Keith, Clearleft

2015年8月

作者简介

本书作者 Aaron Gustafson 是 Web 标准化项目的前项目经理，一直致力于 Web 的标准化和可访问性方面的工作，并且积累了丰富的经验。

在近二十年的 Web 从业生涯中，Aaron 服务过众多我们所耳熟能详的企业，包括 Box、Happy Cog、美国职棒大联盟、McAfee、纽约时报、SAS、StubHub、美国国家环境保护局、Vanguard、Walgreens 和 Yahoo 等。Aaron 亦作为 Web 标准化的顾问参加了微软浏览器团队的研发工作。



Aaron 善于通过写文章来分享他的经验和知识。他为《A List Apart》杂志撰写的“渐进增强三部曲”长期以来备受读者欢迎。他围绕渐进增强这个主题写成的图书《自适应 Web 设计》也获得了非常好的口碑和销量。除了写作，Aaron 现在经常参加各种 Web 大会，与大家分享 Web 技术和设计方面的思考，并在世界各地和当地 Web 从业者进行研讨会形式的交流。

在 Aaron 的家乡，田纳西州的查塔努加市，他建立了查塔努加开放设备实验室。在这个实验室中，Aaron 和 Kelly McCarthy 组织了很多创意和技术的分享活动。Aaron 是 Rosenfeld Media “专家”俱乐部的成员，其博客地址是 aaron-gustafson.com。

译者简介

翟东方，北师大文学系，互联网内容方向从业十年：苏州街角撸过码，西溪湿地画过图，中关村里奋过袖，后厂村口堵过车；曾向往入读北大哲学系，屡屡失利；热衷于用代码解决设计问题，有些心得。梁任公曾曰：“启超没有什么学问……”话锋一转，“还是有些的嘛！”既醉心之，则发展之，则分享之，独乐乐不如众乐乐。Quora 上有云：“Don't waste time on trying to find meaning and purpose in life other than your instincts.（与其努力向别人证明什么，不如跟随自己的天赋做点什么）。”颇以为是。

任洁，北航交互设计硕士。互联网从业九年，走了一条从视觉设计到交互设计再到产品经理的道路，也因此接触了从电商门户到 O2O 等各种各样的产品形态。平时喜欢研究民俗和东亚志怪文献。自幼跟随老师学习工笔白描，热爱一切复杂美丽的线描图案。

最后，欢迎加入读者交流 QQ 群：364021031。

致谢

在一切开始之前，我要声明，这本书的成书离不开导师以及行业中众多朋友和伙伴的帮忙，实际上，如果没有大家的帮助，我一个人是不可能完成这项颇为艰巨的任务的。所以，我必须用一定的篇幅对这些帮助过我的人表示真心的感谢。

首先我要感谢 Molly Holzschlag，还有 Jeffrey Zeldman。感谢你们二位的指导，你们引导我走上了这条写作和分享之路。同时，我也要感谢许多行业大会的组织方，以及出版社的朋友们，给了我表达自己想法的机会。

Steph Troeth 将我头脑中的凌乱思路重新梳理，是她将此书的结构反复调整，以求表达清晰。在她的支持和鼓励下，整个写书的过程变得让人非常享受。

Chris Casciano、Craig Cook，以及 Steve Faulkner，是他们帮助我重新校对了我的文字，指出了我的疏忽之处，确保了内容的简洁清晰。感谢他们卓越的工作和专业精神。

感谢 Tim Kadlec、Jeremy Keith 和 Ethan Marcotte，这几位是行业中的大师级人物了，感谢他们认真读了我这本书的草稿，给出了非常多专业的意见。

Veerle Pieters，感谢她在工作繁忙的间隙依然抽出时间阅读和感受这本书，并为第一版设计了漂亮的封面。

感谢 Ben Dicks 为这本书的排版付出的心血，以及所有这本书中引用的插画的作者的辛勤工作。

感谢 Jeff、Matt、Adam，以及整个 Perma 网页快照维护团队，让我可以放心地在文章中添加超链接。

感谢 Pearson/New Riders 出版集团，其中尤其感谢 Nikki McDonald 对本书的引进，以及 Tracey Croom 和 Mimi Heft 在这本书出版过程中所给予的杰出帮助。

当然，最后要感谢 Kelly，在你的支持下，我才有充足的时间，保持专注来写这本书。因为你的鼓舞，我才有动力完成这项工作。

目录

第1章 为体验而设计 / 1

- 1.1 优秀的代码，麻烦的设备 / 2
- 1.2 早期的 Web 是什么样 / 4
- 1.3 技术实现和用户体验 / 6
 - 一个教训 / 7
- 1.4 你满足不了所有的人 / 9
- 1.5 向后兼容，向前适应 / 10
 - 技术服务于内容 / 11
- 1.6 适应未知的设备 / 12
- 1.7 提升 Web 的可访问性 / 14
- 1.8 分层次设计 / 16
- 1.9 渐进增强的设计思想 / 19

第2章 内容是一切的基础 / 21

- 2.1 让文案和用户产生交流 / 23
- 2.2 设计稿中引入有意义的文案 / 24
- 2.3 优化文案，更好地与用户对话 / 26
 - 问题出现时，做好准备 / 28
- 2.4 文案不确定怎么办 / 30
- 2.5 为用户优化文案内容 / 32
- 2.6 多种形式的内容 / 33
 - 2.6.1 分析成本和收益 / 33

2.6.2 保持内容的可访问性 / 37

2.7 收集数据的时候, 请保持和用户的对话 / 40

2.8 允许留白 / 41

2.9 让内容带路 / 43

第 3 章 使用文档结构化标记增强用户体验 / 45

3.1 那些年的“网页重构” / 47

3.2 利用 HTML 强调内容 / 49

3.3 实现方法不止一种 / 49

避免可能出现的问题 / 52

3.4 如何使用 class 和 id / 54

3.4.1 使用微格式结构化内容 / 56

3.4.2 RDFa 和微数据 (Microdata) / 58

3.5 元素的取舍和选择 / 60

3.5.1 注意文档的大纲结构 / 62

3.5.2 注意代码顺序 / 65

3.5.3 避免加入多余的 HTML 标签 / 69

3.6 使用 ARIA 提升页面可访问性 / 71

3.7 容错处理 / 73

3.8 标记语言对于用户体验的意义 / 77

第 4 章 通过视觉设计提升用户体验 / 79

4.1 设计一个系统, 而非一堆页面 / 80

4.1.1 设计清算 / 81

4.1.2 从设计风格组件开始, 发现整站的视觉语言 / 82

- 4.1.3 建立视觉规范或组件库 / 83
- 4.2 不做无米之炊 / 84
 - 4.2.1 营造和用户的对话场景 / 84
 - 4.2.2 考虑极端情况 / 86
- 4.3 理解 CSS 渲染规则 / 87
 - 4.3.1 就近原则 / 87
 - 4.3.2 权重高比就近更重要 / 90
 - 4.3.3 利用 CSS 容错性 / 92
 - 4.3.4 渐进式导航的实例 / 97
- 4.4 从小屏幕开始的设备响应 / 101
 - 4.4.1 为所有人提供支持，为一些人提供优化 / 105
 - 4.4.2 布局从小到大变化的实例 / 107
 - 4.4.3 允许流动布局 / 109
- 4.5 紧跟标准 / 111
- 4.6 防守性的设计 / 114
 - 保守地应用样式 / 115
- 4.7 在页面中隐藏内容 / 116
 - 4.7.1 避免使用的方法 / 117
 - 4.7.2 可以适当使用的方法 / 118
 - 4.7.3 最佳方法 / 119
- 4.8 使用 CSS 生成内容 / 120
- 4.9 针对其他媒体类型和交互方式而设计 / 122
 - 4.9.1 设计一个供打印的页面 / 123
 - 4.9.2 兼容不同的交互方式 / 124

- 4.9.3 考虑大屏幕设备 / 126
- 4.10 保留浏览器的默认样式 / 128
 - 4.10.1 关于浏览器默认控件是否美观的讨论 / 129
 - 4.10.2 关于模拟系统控件的讨论 / 129
- 4.11 拥抱可持续性 / 130

第 5 章 通过交互设计提升用户体验 / 133

- 5.1 了解潜在的问题，避免日后踩坑 / 135
 - 5.1.1 了解 Web 的性质 / 136
 - 5.1.2 什么情况下会“无 JavaScript” / 138
- 5.2 保证最基础的用户体验 / 140
- 5.3 防止性的代码 / 142
 - 5.3.1 先观察，再动手 / 142
 - 5.3.2 特性检测 / 145
 - 5.3.3 确认代码依赖的库的存在 / 146
- 5.4 建立最低限度的代码支持 / 147
- 5.5 区分对待 / 149
- 5.6 创建页面所需的结构 / 150
- 5.7 描述当前发生了什么 / 152
- 5.8 代码简单明晰 / 153
- 5.9 适应性的界面 / 156
- 5.10 不要提前应用样式 / 159
 - 5.10.1 在 HTML 中添加标记 / 160
 - 5.10.2 充分利用 ARIA / 161
- 5.11 通过适当调整需求提升页面体验 / 161

5.12 鼠标事件之外 / 164

5.12.1 发挥键盘的作用 / 164

5.12.2 提升触屏体验 / 166

5.13 不要依赖网络状况 / 169

5.13.1 将数据存储在手机端 / 169

5.13.2 进一步丰富离线体验 / 171

5.14 合理发挥技术能力 / 172

第6章 持续的手艺活儿 / 175

6.1 体验图 / 176

6.1.1 lx图的优点 / 177

6.1.2 图片延迟加载的实例 / 178

6.1.3 标签式界面实例 / 184

6.2 从历史中学习，寄希望于明天 / 185

6.2.1 移动互联网时代，等价于一个新的拨号上网时代 / 186

6.2.2 以前的小屏幕，变成了现在的大屏幕 / 186

6.2.3 仅有文字的设计方案回归了 / 187

6.3 对任何情况都要做好准备 / 190



第1章

为体验而设计

互联网中，唯一不变的就是变化。四下观望，永远都有新的设计风格，新的语言、框架、工具、新的上网设备在博人眼球。乱花渐欲迷人眼，城头变幻大王旗，连“上网行为”这个概念本身都不能保证难免一天不会被重定义。

在这样一个日新月异的行业，人们普遍感到一种技术焦虑，作为置身互联网行业中的你，我猜也有和我相同的感受。

在我近 20 年的互联网从业经验中，我目睹了太多技术的变迁，从 Java applets、Shockwave、Flash、Prototype、jQuery、960gs、Bootstrap，到 Angular、React，等等，技术的发展就像长江后浪推前浪；在技术发展的同时，设备的屏幕也由小变大，又由大变小，设备的更迭也犹如花开花谢；然后，我们还经历了因设备碎片化导致的开发混乱，面临过为不同设备独立开发，还是一套代码兼容不同设备的概念挣扎。后之视今，亦犹今之视昔。大江东去浪淘尽，唯有 Web 这个概念，在不断发展变化中保留了下来。

Web 的概念之所以经久不衰，是因为它无关设备和屏幕尺寸；Web 不是软件，不需要用户去安装；Web 本身对其承载的内容具有良好的适应性和无限的延展性；Web 可以作为内容访问的窗口，集成在任何需要的地方；Web 可以触及每

个用户；当今的 Web 几乎能够胜任一切。

1.1 优秀的代码，麻烦的设备

2012 年初，我所在的公司接待了一个客户，这个客户被他们移动端 APP 的用户登录验证问题伤透了脑筋，所以想请求我们的帮助。情况是这样的，他们开发了好几个原生 APP，这些 APP 都有一个用户登录的公用模块，是通过 Web Service 的方式访问服务器上的统一接口来实现的。这种技术方案虽然方便了代码的管理，却为他们后来的开发留下了麻烦。他们想在 Web Service 的模块中加入一个安全验证的步骤（类似“你就读的第一所学校的名称是什么？”这种验证问题），却发现部署这个页面比改这个页面麻烦多了，大概步骤如下。

1. 将安全验证这个需求开发完成。
2. 将代码放在 Web Service 上。
3. 更新每个原生 APP，以使新需求生效（可能需要调整每个原生 APP 的 UI）。
4. 将每个 APP 提交到应用市场。
5. 指望用户更新到新版本的 APP。

这位客户解释了他们为什么会采用原生 APP 内嵌 Web Service 的解决方案。他们的 APP 的版本覆盖了 iPhone、iPad 和 Android 三个平台，希望通过 Web 方式统一管理用户登录行为，利用 Web Service 和原生代码分离的特点降低代码耦合，提高重用效率，且做到功能的迅速部署和更新。事实上，这种方式确实已经被大多数网站所采用，并且收到了很好的效果。

了解了客户的困扰和需求后，我们开始了对这个项目的改造。尽管要考虑到跨三个平台的问题，但我们并没有过于纠结每个平台的技术差异和繁杂的屏幕尺寸。我们的思考方式是，设想每个组件在每个屏幕下可能的展示效果。首先，页面的布局必须设计为响应式的，无论是通过小屏幕或大屏幕设备访问，都能保证可见模块有良好的适应性。然后，我们使用了一些轻量的 HTML5 和 JavaScript 技术，希望在尽量不增加文件大小的情况下为页面加入一些新特性（例如使用 HTML5 原生表单验证来代替客户端 JavaScript 验证），这样，在 CSS 或 JavaScript 因网络或者用户设置的原因而无法加载的情况下，页面依然能够验证和响应用户输入的内容。

于是，客户得到了解决方案，高高兴兴地走了。我们也很开心，喝喝酒做点别

的，时间过得很快。几个月后，这个客户又回来了。他们这次开了一个新的脑洞，想让所有访问他们触屏版网站的用户，也能通过之前那套用户验证系统来登录。他们提供了一份触屏版网站的访客 UA 数据（即用户通过何种设备或浏览器访问过这个登录页），数据显示，在两天之内，UA 竟然有 1400 种，就是说，用户通过 1400 种不同的设备和浏览器访问了这个页面。客户拿着这份数据，说兼容问题你们能搞定吗？我们秉着客户至上的精神，认真分析他们提供的数据¹并进行了整理。我们在数据分析结果中发现，这 1400 条登录记录中，有 25 种设备的使用频率极高，差不多覆盖了 97% 的用户。其他 3% 的用户形成了数据长尾，他们使用的设备林林总总，某些设备实在超出我们的想象之外。得到这个数据后，我们感到一阵暗爽，因为在几个月前的开发中，我们已经把 97% 的问题解决啦！所以我们只需要解决其他长尾设备可能出现的问题，方式是——遇见一个解决一个。

客户给我们的预算中，包括兼容 1400 种不同的设备或浏览器的开发成本，包括兼容一些很古老的 Web 设备（例如 Blackberry4 和 Openwave 这种），对这些古老设备的兼容，花去了预算的 1/3。

当所有需求都厘清并完成的时候，我们盘点了我们的项目实际工时和完成时间，大约只花掉了目标预算的 1/2。我们自己很开心，因为我们提前交付了项目，显得我们的团队很高效。同时我们的客户也很开心，因为他为公司节省了很多开发经费，他的老板势必会说，你真能干啊（这些是我们这群不通人情世故的程序员的想象，实际情况我们就知道了）。

一切归功于几个月前，我们对项目的改造，几乎使我们不用做什么，就解决了 97% 的问题。这一个项目的快速推进，靠的不是我们解决 bug 的能力和执行速度，而是印证了项目遵循一个好的设计开发思想是多么重要，这种设计和开发思想，就叫做——渐进增强。

渐进增强是一种拥抱未来的设计思想。这种设计思想无关你面对的是何种设备或浏览器，也无关你的代码是 HTML 或 CSS 的哪个版本。使用渐进增强的设计思想便意味着，你为用户提供了一种无论使用什么设备和浏览器都能正确访问你的页面内容的方法。

听起来很棒啊！你可能会想，这么强大的思想，落实到代码上需要的工作量也很大吧？哈哈，你想错了，一旦你理解了渐进增强思想的实现原理，你会发现

1 作者的分析代码详见 <https://perma.cc/4EAE-Y9H5>

其实落实起来相当简单。

在 2003 年美国西南交互设计大会上，Web 标准化组织的 Steve Champion 首次提出了“渐进增强”的概念，用来描述一种 Web 设计的思考角度——即围绕页面中的内容，从内容出发进行设计。一旦你明白了渐进增强这个概念的内涵，你就会不由自主地顺从某种逻辑来思考。谢天谢地，在 Web 诞生差不多 10 年之后，我们终于有了一套 Web 设计的指导思想¹。

1.2 早期的 Web 是什么样

最初，我们的电脑是没有图形界面的²，我们看到的，只是黑色屏幕上的一行行绿色的命令行和文字³（图 1-1），上网的用户个个看起来都和程序员没什么区别。

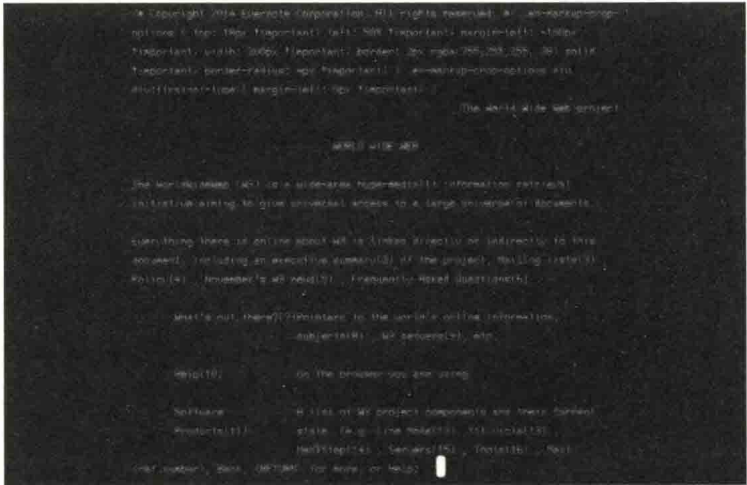


图 1-1 2013 年版的命令行模式浏览器

命令行模式的浏览器仅支持一些基本的文本格式，如缩进、对齐等，但是在 1990 年，Web 上传输的内容基本都是些文本和用于发表的内容，所以这样的内容渲染程度也是能够满足人们的需要的。

1 本书成书的目的即把渐进增强的思想介绍给大家。
2 最早的图形浏览器是一款叫做 World Wide Web（后因与万维网重名，改名为 Nexus）的产品。但是因为它只能运行在 NeXT 操作系统上，所以并未广为人知。
3 我的一些朋友和同事在 2013 年的时候回到 World Wide Web 浏览器的诞生地——欧洲核子研究中心（CERN），开发了一款使用现代浏览器内核技术的命令行浏览器，并撰写了相关文章。感兴趣的话点击[这里](http://perma.cc/2UYR-HVWP)：<http://perma.cc/2UYR-HVWP>