

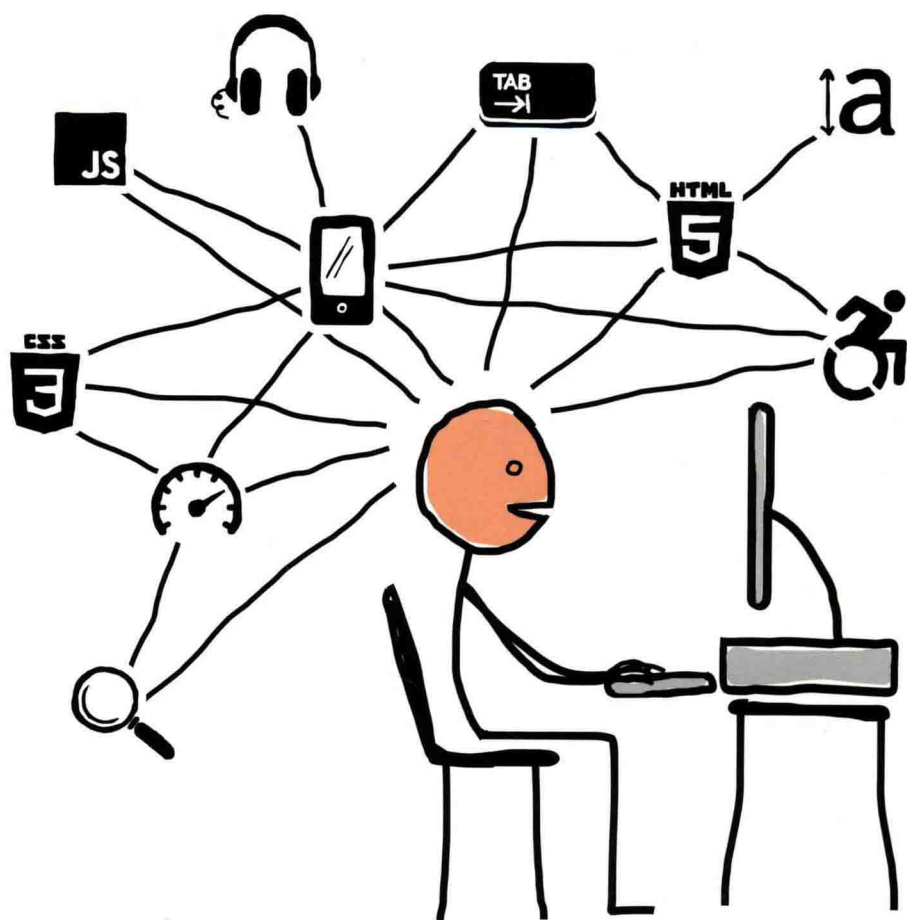
中国信息无障碍产品联盟推荐图书

包容性Web设计

INCLUSIVE DESIGN PATTERNS

Coding Accessibility Into Web Design

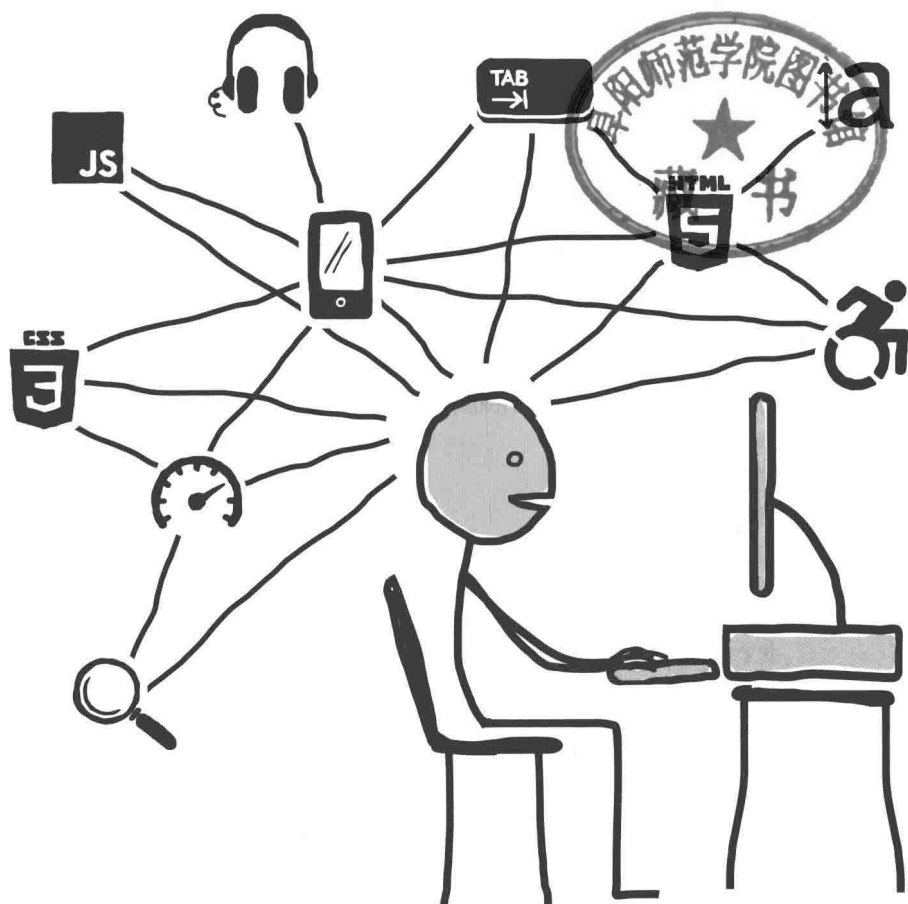
[美] Heydon Pickering 著 于坤 译



包容性Web设计

INCLUSIVE DESIGN PATTERNS
Coding Accessibility Into Web Design

[美] Heydon Pickering 著 于坤 译



人民邮电出版社
北京

图书在版编目 (C I P) 数据

包容性Web设计 / (美) 海顿·皮克林
(Heydon Pickering) 著 ; 于坤译. — 北京 : 人民邮电
出版社, 2017. 7
ISBN 978-7-115-45818-6

I. ①包… II. ①海… ②于… III. ①网站—设计
IV. ①TP393.092.2

中国版本图书馆CIP数据核字(2017)第133574号

版权声明

Inclusive Design Patterns

978-3-945749-43-2

Heydon Pickering

Copyright ©2016. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, by photocopying, recording or otherwise, without the prior permission in writing from Smashing Media GmbH. CHINESE SIMPLIFIED language edition published by POSTS & TELECOM PRESS.

本书中文简体版由 Smashing Magazine 公司授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。版权所有, 侵权必究。

◆ 著 [美] Heydon Pickering

译 于 坤

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>

北京画中华印刷有限公司印刷

◆ 开本: 720×960 1/16

印张: 11

字数: 192 千字

2017 年 7 月第 1 版

印数: 1—2 000 册

2017 年 7 月北京第 1 次印刷

著作权合同登记号 图字: 01-2016-9367 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

广告经营许可证: 京东工商广登字 20170147 号

内容提要

前端开发者往往不自觉地就做出可访问性很差的网站，这不是粗心也不是能力问题，而是没有掌握正确的方法，且对当下多平台开发心存畏惧。

本书帮助 Web 设计师、开发者和测试小组，创新性地从设计模式的角度思考可用性体验；帮助开发者破除迷思，澄清误解，以高效的方式制作可访问的界面。此外，本书还介绍了一些前端设计模式来创建包容性体验。

本书思路新颖，内容深入浅出，汇集了作者多年积累的宝贵经验，无论是工作多年的资深前端开发、测试工程师，还是刚入门的前端新人，都能从本书中得到启发。

献给我的妻子 Fan，不管世事变幻，是你让我微笑面对。

作者简介

Heydon Pickering 是一位实用主义设计师、作家和公众演说家，Smashing Magazine 的无障碍编辑，也是 Paciello Group 的顾问。他对使用新的和创新型的方式让网络更具包容性很感兴趣。用户研究、系统思考和古老的语义化 HTML 各司其职。当他不写作、编码、画图时，会参与心理健康运动、声音设计实验，用他失真的电吉他弹奏金属摇滚。当然，也离不开啤酒。

审稿人简介

Rodney Rehm 是一位住在德国南部的网站开发者，拥有 10 年的全栈开发经验，目前在 Deutsche Telekom 的智能家庭平台 Qivicon 做前端工作。他创建了 URI.js 和 ally.js，使 libsass 能在浏览器中运行，并写出了世界上第一个 buggyfill。

Steve Faulkner 是 TPG 欧洲的 Web 无障碍高级顾问和技术总监，于 2006 年加入 Paciello Group，之前是 Vision Australia 的网络无障碍高级顾问。他是 Web Accessibility Toolbar 无障碍测试工具的创建者和首席开发者。Steve 是几个小组的成员，包括 W3C Web 平台工作组和 W3C ARIA 工作组。他还是 W3C 几个规范的编辑，包括：

- HTML 5.1
- ARIA in HTML
- Notes on Using ARIA in HTML
- HTML5: Techniques for useful text alternatives

HTML5accessibility.com 网站也是 Steve 负责开发和维护的。

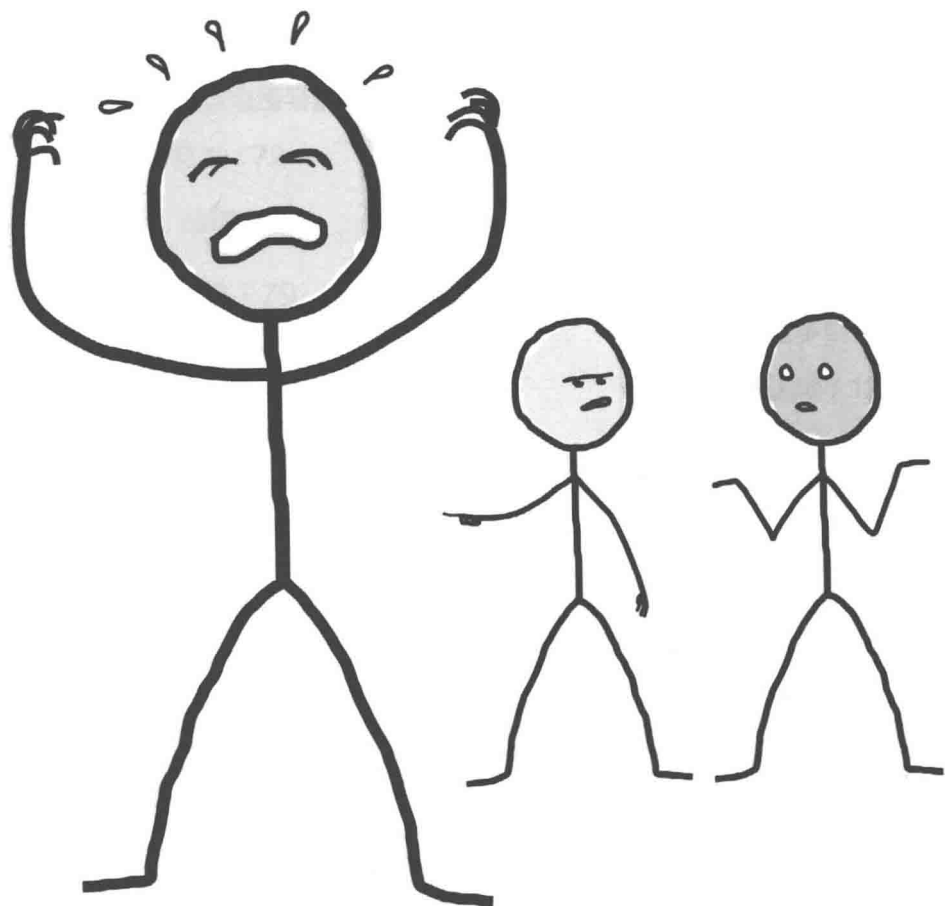
中文版审稿人简介

刘彪是信息无障碍研究会的技术主管，他拥有丰富的信息无障碍实践经验，是中国屈指可数的信息无障碍专家。

加入信息无障碍研究会以来，刘彪为包括腾讯、阿里巴巴、百度、搜狗、滴滴等在内的众多知名公司的产品提供专业的信息无障碍相关服务。

刘彪精通计算机技术，拥有超过 10 年的 C/C++、PHP 经验，曾独立研发 PC 端屏幕阅读软件、构建面向障碍用户基于云的软件市场、创建并运营障碍群体辅助技术在线社区，他还参与了 WCAG2.0 中文版审阅。

我是一名设计师！



这家伙怎么了？

不知道，我只是问他厕所在哪儿。

目录

第 1 章 引言 / 1

- 1.1 重新思考 Web 设计 / 2
- 1.2 包容性设计 / 2
- 1.3 包容性的按钮 / 4

第 2 章 文档 / 9

- 2.1 文档类型 (Doctype) / 9
- 2.2 语言属性 (lang) / 10
- 2.3 响应式设计 / 11
- 2.4 字体大小 (font size) / 13
- 2.5 渐进增强 / 15
- 2.6 管理资源 / 16
- 2.7 字体子集 / 17
- 2.8 <title> 元素 / 18
- 2.9 <main> 元素 / 18
- 2.10 把页面组合到一起 / 20
- 2.11 有关框架、预处理器和任务处理器 / 21

第 3 章 段落 / 23

- 3.1 字体 / 23
- 3.2 排版 / 25
- 3.3 行距 (行高) / 27
- 3.4 行内链接 / 28
- 3.5 自动加图标 / 31
- 3.6 写段落 / 32
- 3.7 总结 / 33

第 4 章 博客文章 / 35

- 4.1 <main> 元素 / 35
- 4.2 标题结构 / 36
- 4.3 <article> 元素 / 41
- 4.4 渐进增强与互用性 / 43
- 4.5 标题和链接文本 / 45
- 4.6 视频 / 46
- 4.7 使用流式布局 / 50
- 4.8 总结 / 53

第 5 章 按模式评估 / 54

- 5.1 按规则评估的问题 / 54

5.2 自定义元素和 Shadow DOM / 58	9.3 产品缩略图 / 103
第 6 章 导航区域 / 63	9.4 “立即购买” 操作 / 106
6.1 导航地标 (navigation landmark) / 63	9.5 总结 / 113
6.2 站点范围导航 / 64	第 10 章 筛选部件 / 117
6.3 目录 / 72	10.1 标签结构 / 117
6.4 总结 / 77	10.2 CSS 增强 / 119
第 7 章 菜单按钮 / 79	10.3 JavaScript 增强 / 121
7.1 外观 / 79	10.4 加载更多结果 / 124
7.2 渲染图标 / 80	10.5 显示选项 / 127
7.3 文本标签 / 84	10.6 允许动态 内容 / 132
7.4 旧版浏览器 / 85	10.7 总结 / 133
7.5 操作 / 86	第 11 章 注册表单 / 137
7.6 触摸目标 / 89	11.1 表单的上 下文 / 137
7.7 总结 / 90	11.2 基本表单 / 139
第 8 章 包容性的原型 / 92	11.3 必填字段 / 143
8.1 纸质原型 / 92	11.4 显示密码 / 144
8.2 从纸到代码 / 96	11.5 表单验证 / 145
第 9 章 产品列表 / 99	11.6 微文案 / 150
9.1 列表的优点 / 100	11.7 总结 / 151
9.2 关键信息 / 102	

第 12 章 测试驱动的 HTML

标签 / 153

12.1 选择器逻辑 / 154

12.2 测试驱动器的选项

卡界面 / 154

12.3 并非放之四海而

皆准 / 162

第1章 引言

假设我们在一个聚会上见面了，几句寒暄后，你问我是做什么的，“我是个设计师”我回答道。你还没有来得及问我是什么设计师，我们共同的朋友——聚会主人——拍我的手肘说：

“能占用你点时间吗？”

主人向你说了声抱歉后便带我离开了。你松了一口气，因为那种自豪地宣称自己是“设计师”的，很可能是一个很自负的人。尽管如此，你还是好奇我到底是做什么的。也许在设计杯子？缝制后现代主义的泳装？开发直升机导弹系统？可以指任何事情。

不过直到聚会结束，我们也没有机会重新认识。第二天与我们共同的朋友打电话时，他再次为前一天打断你致歉。过了一会儿，你回忆起来。

“哦，没关系。你的设计师朋友，具体设计什么？”

“呃，他是个 Web 设计师。”

“哦，这样啊。”

“是啊。”

现在，你的反应取决于你对庞大、模糊的公共信息母体（我们称之为 Web）的关系。如果你只是互联网的消费者，而不是生产者，知道“Web 设计师”这个词就足够了：“哦，他设计网络上的东西，网站之类的。

我打赌你不是这类人，否则你不会读这本书。这时候你明白，在互联网行业中“设计师”一词多么有争议，甚至与经典设计术语的定义截然不同。所以哪些人是设计师，他们具体做什么？

很长一段时间以来，这个问题的答案都是严重错误的。“设计师”一词被误解，给从业者平添了不少麻烦。

1.1 重新思考 Web 设计

我们这个行业最根深蒂固的错误就是，将印刷介质的视觉传达设计中的原理应用于 Web。印刷设计师的领域纯粹是视觉的，局限于可预测的空间，使用商定好的材料，可以还原设计稿中的分辨率和颜色。印刷设计的成品是静态的，不能变。

显然 Web 不是这样的，但人们还是花费巨大的精力以期许毫无意义的像素级完美，还将我们从业者武断地划分为设计师和开发者，而没有考虑到还有内容编辑和项目经理！

实际上，设计师的工作应当深思熟虑、仔细斟酌，对给定的问题追求最佳解决方案。如果把设计仅仅局限于视觉美学，会让 Web 变得杂乱无章，更别提会导致网站无法访问、性能低下，当然实用性也会很差。

本书的目的是培养你的 Web 设计思维，并关注一些（令人惊讶的是最近才有的）Web 设计学科的发展。

- 网页是由代码组成的，因此，用代码设计才是用对了材料。这是最好的实践方案。
- （我们通过网络写出来，或以其他方式表达出的）内容，必须遵循设计思维，事实上，所有其他设计决策都应该有助于实现这个目的。
- 网页不是一成不变的，应该支持变化的动态内容。这些内容应该用组件管理，组件可以用约定好的模式重复使用。
- 网站或 APP 的潜在受众是所有人，Web 的包容能力、用户的偏好和使用环境是最重要的。受众不同（总是不同）时，包容性的界面才是最可靠的。

1.2 包容性设计

本书的主题和目的，最后也是重要的原则，在我们研究包容性模式之前，首先必须定义包容性设计。它更像是一种心态，而不是可以简化的技能，所以我要通过一个类比来说明。

以下的 JSON 代表了一个街道地址：

```
"address": "84, Beacon St, Boston, MA 02108,
United States"
```

现在如果我想使用像 Handlebars 这样的模板库，来打印地址到屏幕，我可能会这样写模板 `{{this.address}}`，但是如果我需要用地址中的国家怎么办？当前的数据形式是字符串，我必须写一个帮助函数取出“United States”这部分。

```
Handlebars.registerHelper('getCountry', function(address) {
  return address.split(',').pop();
});
```

这其实是一种 hack：一种脆弱又复杂的解决方法。出现了 hack 就是坏设计的征兆。使用辅助函数不仅计算量大，也不可靠，为什么？因为并非所有地址都以国家 / 地区结尾，例如英国地址通常以邮政编码结尾：

```
"address": "85-87, Gwydir St, Cambridge, England, CB1 2LG"
```

一个更健壮的解决方案是，让地址属性成为一个对象，并将地址的每个部分存储为该对象的一个属性：

```
"address": {
  "building": "85-87",
  "street": "Gwydir St",
  "city": "Cambridge",
  "country": "UK",
  "zipOrPostcode": "CB1 2LG"
}
```

现在我们可以简单地用 `address.country` 取得国家。

这似乎是一个更好的方法，但真的是这样吗？使用一个巨大的国际地址集合，你会发现它们区别太大，不可能规定一个固定的属性集。简单地截取字符串更合适一些，但有时候也不能获取到国家。

无论如何，考虑数据结构试图获得一个最佳解决方案就是设计。不需要买 Adobe 软件，也不用 Sketch。实际上这就是一种包容性设计：正确的解决方案是包容各种不同类型的地址。我们希望能处理一致、统一的地址，但这不现实，所以我们也考虑例外情况。

上一段的句子中，用“界面”替换掉“数据的结构”，用“人”替换掉“地址”，你就理解包容性设计了。

最棒的就是，设计包容性的界面就像设计健壮的数据结构一样，没有变得更难、更复杂，只是不一样而已。本书将通过包容性的透镜查看常见的 Web 界面，帮你快速了解如何运用和重复运用包容性设计的约定，从而给你带来更广泛、更忠实的受众群体。

1.3 包容性的按钮

来看一个更简单的例子，一个界面模式的原型很像这样，我们从 3 种类型的设计师角度看待一个交互元素——按钮。这个例子的目的是向你展示：多了解一点这方面的知识，就可以做出更简单、更具包容性的方案。

1.3.1 视觉设计师

第一个设计师有平面设计背景。他们的文件从来没有低于 300dpi，他们懂得颜色理论，排版作图的技巧能力也很强。对他们来说，按钮是一个视觉工艺品，可以在 Adobe Illustrator 或 Sketch 中制作。他们关心的是，如何将这个按钮做得像一个真正的按钮，同时还要适应品牌。他们不知道如何把这个按钮放到网页上或者让按钮做其他动作。

1.3.2 写代码的设计师

第二个设计师会很多和第一个设计师一样的技能，但在一个重要的方面不同：他有充足的 HTML、CSS 和 JavaScript 知识，能把按钮放到网页中，并附加一个 JavaScript 监听事件。

HTML 结构看起来是这样的：

```
<div class="button"></div>
```

CSS 是这样的：

```
.button {  
  width: 200px;  
  height: 70px;  
  background: url('../images/button.png');  
}
```

JavaScript 可能是用 jQuery 或者 Angular JS API 写的，也可能是用 Web API（原生 Javascript），可能看起来是这样的：

```
button.addEventListener('click', function() {  
    // the event fired on click  
});
```

虽然被称为设计师，但是他已经学会了编码，可以把想法放到 Web ——把它们放到网页中，甚至某些功能对小部分用户还可用。但不幸的是，除了监听事件，他们真正做的是编码还原视觉设计，而不是用代码设计。第三个设计师已经意识到这点，平面设计和 Web 界面设计几乎没有直接的转换关系。

1.3.3 包容性设计师

第三个设计师从多个角度看待第二个设计师的按钮，每一点都是站在一个潜在用户的角度，能看到当前做法导致的许多问题。

一个问题是，缩放网页以方便阅读的用户不少，但如果用户整页缩放，而网页中的图片又不是矢量图，那就无法做到缩放不失真。如果用户只调整了浏览器的默认字体大小，则图片（因为使用的是像素单位而不是相对单位）将不会缩放。

另外一个问题是，当用户使用移动网络，关闭图像以节省流量时，完全由背景图像构成的按钮将不可见。用户打开高对比度模式的时候，有时候很难区分前景和背景图像，在某些情况下，也会看不到背景图。

不止于此，<div> 元素跟内建的 <button> 元素不同，它在当前的表单中没有获取焦点或者由键盘操作的能力。有些人喜欢用键盘导航和操作网页。有些人却只能用键盘，因为鼠标需要更精细的手指动作控制，他们做不到。

屏幕阅读器用户、视力严重受损的人，还有通过使用合成语音读出网页以辅助理解的人，他们的访问能力都被剥夺了。大多数桌面屏幕阅读器用户也是键盘用户，所以他们会遇到同样的问题，此外 <div> 没有语义含义，不能提供非视觉的按钮信息。

由于按钮的“Start”文字在背景图像中，（为残疾人设计的）辅助技术也无法起作用。同样的原因，按钮也不能被翻译成其他语言，它也限制了国际受众，错失了很多用户。