

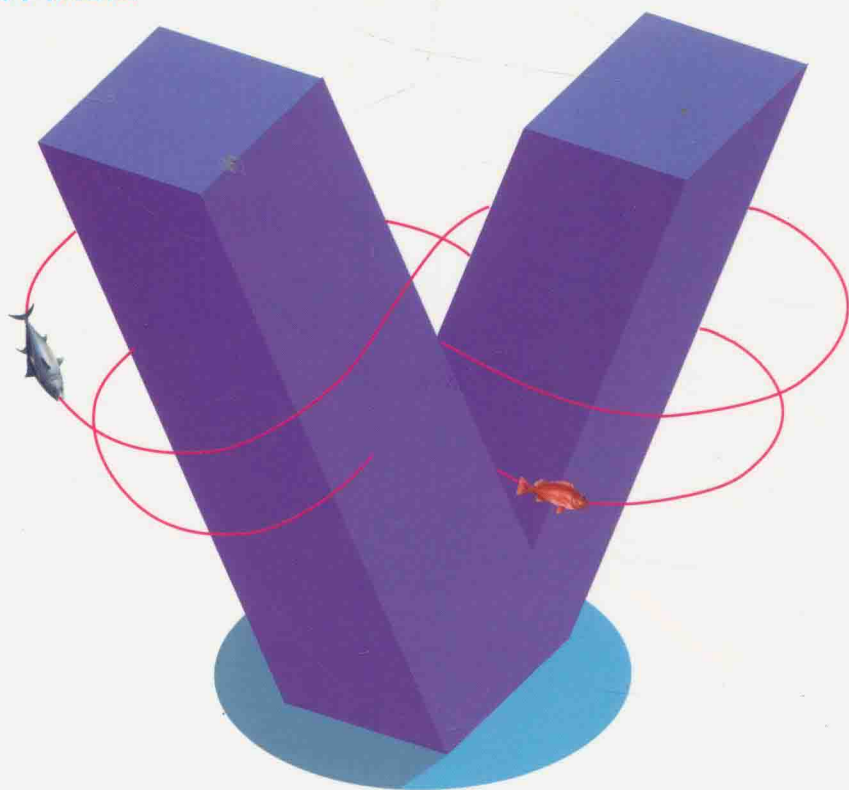
RWD

响应式 网页设计实战

Bootstrap
Foundation
Pure 的
学习与使用

丰富的范例程序和详细的图解
逐一介绍响应式网页设计的
核心技术和方法

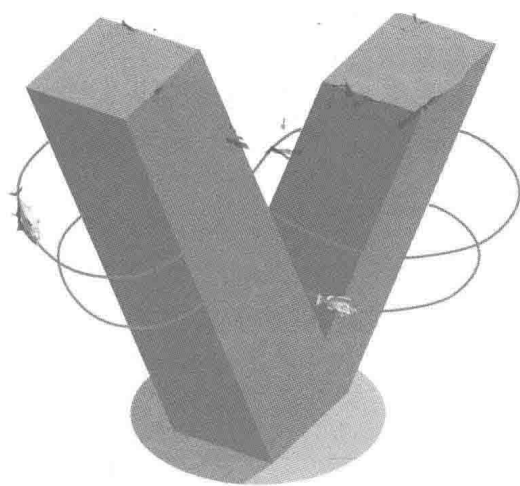
柯霖廷 等编著



机械工业出版社
China Machine Press

响应式 网页设计实战

柯霖廷 等编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

响应式网页设计实战 / 柯霖廷等编著. —北京: 机械工业出版社, 2016.10

ISBN 978-7-111-54942-0

I. ①响… II. ①柯… III. ①网页制作工具 IV. ①TP393.092.2

中国版本图书馆CIP数据核字 (2016) 第231156号

自适应/响应式网页设计 (RWD) 就是集中创建页面的图片排版大小, 智能地根据用户行为以及使用的设备环境 (系统平台、屏幕尺寸、屏幕定向等) 进行相应的布局。本书以丰富的范例程序和详细的图解逐一介绍响应式网页设计的技术架构及设计要领, 使用HTML与CSS设计网站的基本策略与技术, 以及响应式网页设计的模块应用, 具体内容包括自适应/响应式网页设计概述、网页的新旧切版方式和字体资源、版面尺寸的固定方式与弹性方式、Bootstrap 的学习与使用、Foundation 的学习与使用、Pure 的学习与使用。

本书内容由浅入深, 同时辅以易于测试和操作范例的程序, 便于读者详细了解程序的运行原理, 是一本很好的程序设计参考书。

响应式网页设计实战

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: 夏非彼 迟振春

印刷: 中国电影出版社印刷厂

开本: 188mm × 260mm 1/16

书号: ISBN 978-7-111-54942-0

版次: 2016 年 10 月第 1 版第 1 次印刷

印张: 15

定价: 49.00 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) 88379426 88361066

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 88379604

读者信箱: hzit@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



序 1

响应式网页设计就是本书的核心。在智能手机、平板电脑等移动设备大行其道的时候，网页设计人员如果不会灵活和熟练地运用响应式网页设计技术将电脑桌面网页设计技术拓展到移动设备上，那么不但自己会疲于奔命，而且最终会被市场淘汰。

本书以丰富的范例程序和详细的图解逐一讲解响应式网页设计的核心技术和方法，这种网页设计技术和方法会根据用户所使用设备的浏览器环境（例如屏幕的长度、宽度、长宽比、分辨率或设备屏幕显示的方向等）自动调整网页的版面，将恰当的内容和最佳的显示结果提供给用户。简而言之，就是网页设计人员只要设计统一版本的网页程序，就能在电脑、智能手机和平板电脑等各种设备上完整展示网页内容，而无需为不同屏幕大小和功能的设备分别设计和改写网页程序。

本书涉及的内容包括HTML4、HTML5、JavaScript和CSS，读者需要具备一些相关方面的知识，才能在学习时事半功倍。

HTML5 标准相对比较新，并不是所有的浏览器或者它们的旧版本都能完整地支持本书中含有HTML5 语言功能的部分范例程序。要正确地运行这些范例程序，建议使用最新版本的浏览器，本书的所有范例程序都在Chrome 版本 47（谷歌的浏览器）上调试并通过。

当然，支持HTML5 的浏览器还有Windows 10 中的Microsoft Edge、Internet Explorer 11、Opera 版本 33.0 以上以及Firefox（火狐浏览器）和Safari等的最新版本。另外，还有国内的 QQ浏览器、猎豹浏览器等的新版本。如果读者想在这些浏览器上运行 HTML5 的程序，请先阅读版本说明中对 HTML5 支持程度的细节说明。

赵 军

2016 年 8 月

序 2

本书与业界其他相关书的不同之处在于,其各章节涉及的内容都和响应式网页设计的技术架构有关。第 1 章是关于自适应 / 响应式网页设计的概述,第 2 章介绍网页新旧切版的方式和字体资源,第 3 章介绍版型尺寸的固定与弹性,第 4 章介绍 Bootstrap 的学习与使用,第 5 章介绍 Zurb's Foundation 的学习与使用,第 6 章介绍 Yahoo's Pure 的学习与使用。

为了能更好地理解第 2 章开始的各个范例程序,读者需要具备 HTML、CSS 和 JavaScript 的相应知识。若没有这些基础知识,而试图直接通过各个章节的内容来理解各个范例程序,则势必会事倍功半、劳心不已……希望没有基础的初学者,先去学习上述的基础知识,有牢固的根基才好!

从第 4 章开始的内容着重于更为全面的响应式网页设计模块的应用,以指导程序设计人员更快速地创造出具有特定质感和风格的各个组件。各个组件所对应的模块原则上都被打包在外部 CSS 文件和 JS 文件中,并通过 HTML 语句在特定网页中显示出各个组件的外观。

第 4 章至第 6 章分别介绍不同厂商免费提供的组件模块,可以通过 CSS 文件和 JS 文件的链接语句,使得隶属于不同厂商的各个组件模块同时应用于同一个网页当中,再经由美编人员尽量使此网页里的各个组件模块呈现出同一质感和风格,以营造出网页中统一的整体外观。

本书章节内容的安排经由作者巧思而成,以便读者能够轻松驾驭,逐步理解本书的精髓,同时辅以易于测试和操作的范例程序,让读者详细了解程序的运行原理。相对而言,第 4 章到第 6 章的一些范例程序有些烦琐与复杂,因此以特定的程序代码区段为单位来测试其显示的组件外观,是一种较为惬意的学习方式。

本书是一本程序设计参考书,读者若觉得一大段程序代码让自己感到心烦且有压力感,不妨先看懂特定程序代码所对应的说明文字,并对自己不熟悉的英文单词做个记号,通过在线英文字典进行查询,以加快自己对程序代码的深度理解。

随着时间的推移,本书所用到的各厂商的组件模块也会改版,因此在网络云盘中提供了各章的范例程序文件,下载地址为<http://pan.baidu.com/s/1hsuGdNI>(注意区分字母的大小写以及数字和字母)。如果下载有问题,请电子邮件联系 booksaga@126.com,邮件主题为“响应式网页设计实战”。

在此恭祝各位读者在学习上百尺竿头更进一步,不懈努力,持之以恒,蒸蒸日上,事半功倍!

编 者

目 录

序

第 1 章	自适应 / 响应式网页设计概述	1
1.1	弹性的样式与版面	1
1.2	以移动设备为优先	1
1.3	业界流行的 RWD Framework	2
1.4	并非一定需要 RWD Framework	2
1.5	RWD 实现的机制	5
1.6	浏览器的支持度	5
第 2 章	网页的新旧切版方式和字体资源	7
2.1	程序代码的编辑器	7
2.2	CSS 搭配 HTML4 切版方式	7
2.2.1	开始切版前的 HTML 基本程序代码结构	7
2.2.2	规划要创建的版面	8
2.2.3	建立区块的程序代码	9
2.2.4	标示区块的范围	9
2.2.5	设置区块尺寸	10
2.2.6	限定版面总宽度	12
2.2.7	设置区块为浮动排列	12
2.2.8	调整各区块的间距	14
2.2.9	版面的居中对齐	15
2.2.10	改用百分比作为显示度量单位	16
2.3	CSS 搭配 HTML5 切版方式	18
2.3.1	建立区块的程序代码	20
2.3.2	标示区块范围	20
2.3.3	设置区块尺寸	21
2.3.4	限定版面的总宽度	23
2.3.5	设置区块为浮动排列	24
2.3.6	调整各区块的间距	25
2.3.7	版面的居中对齐	27
2.3.8	改用以百分比为显示度量单位	28
2.4	CSS 搭配 JavaScript 切版方式	30
2.4.1	精简原有的程序代码	30
2.4.2	注释掉整段 CSS 规则语句	32
2.4.3	用 JavaScript 程序代码实现特定版型	32
2.4.4	JavaScript 版型程序代码的第 1 阶段简化	33

2.4.5	JavaScript 版型程序代码的第 2 阶段简化	34
2.5	跨网域运用在线字体	35
2.6	使用本地的字体	38
第 3 章	版型尺寸的固定与弹性	42
3.1	HTML 传统方式	42
3.2	CSS 应用方式	45
3.2.1	@media 语句	45
3.2.2	meta viewport 语句	51
3.3	JavaScript 搭建方式	52
第 4 章	Bootstrap 的学习与使用	56
4.1	Twitter 的 Bootstrap 简易应用方式	56
4.2	免费小图标	58
4.3	下拉式或上拉式菜单	59
4.3.1	下拉式菜单	59
4.3.2	上拉式菜单	61
4.3.3	具有选项分隔线的下拉式菜单	61
4.3.4	具有选项分类和选项分隔线的下拉式菜单	62
4.4	一般按钮和按钮组	64
4.4.1	按钮组	64
4.4.2	按钮组的工具栏	64
4.5	带有菜单按钮的按钮组	65
4.5.1	横向按钮组	65
4.5.2	纵向按钮组	66
4.5.3	窗口宽度型	67
4.5.4	分离型按钮组	68
4.6	不同尺寸的按钮	69
4.6.1	大尺寸按钮	69
4.6.2	标准尺寸的上拉式菜单按钮	70
4.6.3	小尺寸按钮	71
4.6.4	特小尺寸按钮	72
4.7	窗体中的文本栏	73
4.7.1	标签文字组合式	74
4.7.2	单选按钮和多选按钮组合式	75
4.7.3	一般按钮组合式	76
4.7.4	下拉式菜单按钮组合式	77
4.8	窗体中的导航栏	79
4.8.1	默认的标签型导航栏	79
4.8.2	简易标签型导航栏	80
4.8.3	具有窗口宽度的堆栈式导航栏	80

4.8.4	均分窗口宽度的标签型导航栏	80
4.8.5	动态置顶导航栏	82
4.9	浏览分层提示和分页	84
4.9.1	浏览分层提示	84
4.9.2	分页导航栏	85
4.9.3	前一页和下一页按钮	86
4.10	文字标签、文字徽章与文字框	87
4.10.1	文字标签	87
4.10.2	文字徽章	87
4.10.3	文字框	88
4.11	缩略图与图解说明	89
4.12	警告文字框	91
4.13	进度条	92
4.14	媒体对象	95
4.14.1	一般图片列表	95
4.14.2	多层次图片列表	97
4.15	列表分组	100
4.15.1	默认的列表分组	100
4.15.2	加上数值徽章的列表分组	101
4.15.3	按钮型列表分组	102
4.15.4	应用颜色样式的列表分组	103
4.15.5	带有内容的列表分组	104
4.16	面板	105
4.16.1	一般面板	105
4.16.2	带有表格内容的面板	107
4.16.3	带有列表分组内容的面板	109
4.17	自适应嵌入	110
第 5 章	Zurb's Foundation 的学习与使用	112
5.1	Zurb's Foundation 简易应用方式	112
5.2	滑入菜单	113
5.2.1	默认的滑入菜单	113
5.2.2	双边滑入菜单	114
5.2.3	双边多层次滑入菜单	118
5.3	置顶导航栏	122
5.4	工具栏与内置小图标	125
5.4.1	跨窗口宽度的工具栏	125
5.4.2	纵向工具栏	126
5.4.3	内置图标工具按钮所构成的工具栏	127
5.5	侧边导航栏	128

5.6 动态置顶追踪型导航栏	129
5.7 子导航栏	131
5.8 浏览分层提示	132
5.9 分页导航栏	133
5.10 缩略图	134
5.11 缩放式视频	137
5.12 窗体组件	138
5.13 切换按钮	148
5.14 滑杆	151
5.15 窗体验证	156
5.16 按钮	160
5.17 按钮组	162
5.18 分割下拉式菜单按钮组	167
5.19 整体下拉式菜单按钮	169
5.20 字体样式	171
5.21 行内列表	173
5.22 文字标签	173
5.23 modal 窗格	174
5.24 警告框	180
5.25 面板框	182
5.26 动态提示框	183
5.27 页面操作展示	185
5.28 简易型下拉式菜单与下拉式内容框	189
5.29 价目表	192
5.30 进度条	193
5.31 表格	194
5.32 折叠式面板	195
5.33 标签面板	203
5.34 均分版型	206

第 6 章 Yahoo's Pure 的学习与使用..... 210

6.1 Yahoo's Pure 简易应用方式	210
6.2 版型网格和最小容器单元	210
6.3 窗体组件	213
6.4 按钮	222
6.5 表格	224
6.6 菜单	229

第 1 章 自适应 / 响应式网页设计概述

自适应 / 响应式网页设计（RWD, responsive web design）就是指在多种设备的不同尺寸的屏幕上，精巧设计和制作出最佳视觉体验的网页。所谓的最佳视觉体验，是指：

- 用户更容易浏览网页内容。
- 用户需要更少的窗口缩放、平移与滚动的操作。

1.1 弹性的样式与版面

按RWD 方式设计出来的网站，通过可按比例调整的窗格、多媒体对象（图片、视频、动画）与 CSS3 媒体查询语句，使得各个网页的整体版面可缩放自如地显示于各种视觉环境中。举例来说：

- 可按比例调整的窗格，要求在网页中的各元素（文字、窗体、多媒体等对象）都以相对的百分比（%）来表示，而不是以像素（px, pixel）或点数（pt, point）作为尺寸单位。
- CSS3 媒体查询语句允许特定网页根据不同设备的屏幕尺寸或其他特性而应用不同的 CSS 样式规则。

1.2 以移动设备为优先

以移动设备为优先、低调的 JavaScript 与渐进强化支持RWD的方式是RWD机制在形成具体概念之前的基本意识。较早的移动设备上的浏览器并不能解读 JavaScript 语言或 CSS3 媒体查询语句，所以在早期业界较常做的方法是：对于同样内容的网页，为了支持移动设备的用户群，需要再额外制作内容大致相同的网页。

较先进的移动设备必须渐进地支持 JavaScript 语言与 CSS3 媒体查询的新语句，才能使用渐进强化的 RWD 机制。

1.3 业界流行的 RWD Framework

现在业界开始流行所谓的免费 RWD Framework（框架），可大幅减少程序设计人员在编写程序代码所花费的时间，因为这些框架已经内置了以 CSS 和 JavaScript 程序代码为基础的表格、窗体、菜单、按钮、导航栏、文本栏等元素和多种版面类型（简称版型）。

下面给出 3 个知名 RWD Framework 的官方网址。

- Twitter's Bootstrap: <http://getbootstrap.com>, 如图 1-1 所示。

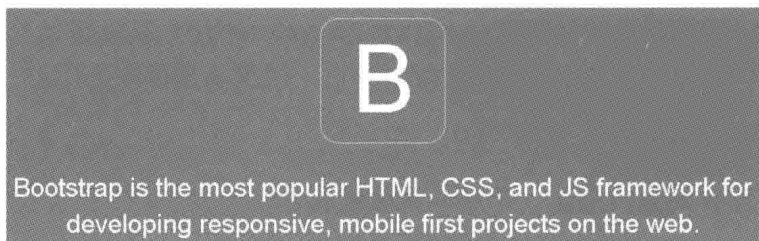


图 1-1 Bootstrap 官网上的文字介绍

- Zurb's Foundation: <http://foundation.zurb.com>, 如图 1-2 所示。

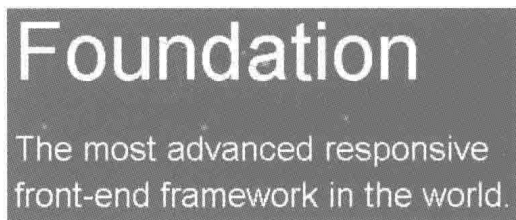


图 1-2 Foundation 官网上的文字介绍

- Yahoo's Pure: <http://purecss.io>, 如图 1-3 所示。



A set of small, responsive CSS modules that
you can use in every web project.

图 1-3 Yahoo's Pure 官网上的文字介绍

1.4 并非一定需要 RWD Framework

程序设计人员要采用 RWD 机制，也不一定要使用各个厂商提供的 RWD Framework。只要充分熟悉 HTML4、HTML5、CSS2、CSS3 与 JavaScript 相关语句，也可以设计出具有独特 RWD 机制的网页。设计实现具有良好 RWD 机制的网页，大致需要具备以下技术（见图 1-4）。

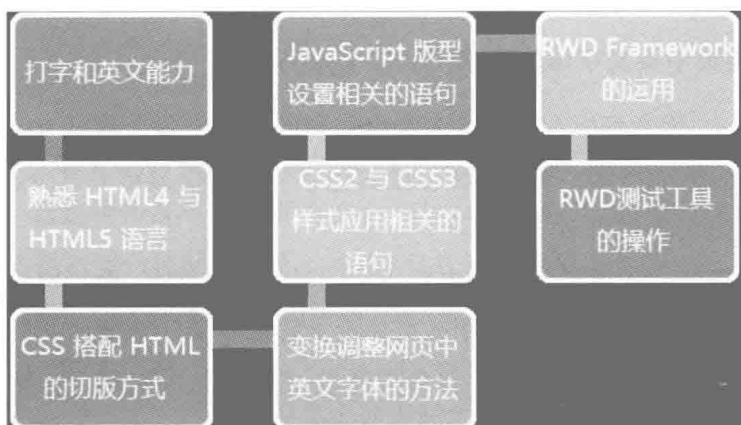


图 1-4 要设计实现 RWD 机制的网页需要具备的技术

1. 打字和英文能力

根据笔者近 10 年以来的教学经验，程序设计初学者的英文能力通常一般，可偏偏任何一种程序设计语言，无论是关键字、保留字还是内置的变量和函数等的命名，都大量采用英文！

在这种情况下，英文水平一般的初学者，在编写程序代码的过程中，会不断地拼错各种名称的英文单词，甚至在大小写字母上出错，导致其程序代码无法正常运行。

笔者也常常遇到一些初学者，一边对比讲师所提供的“相当简短”的范例程序代码，一边检查他们自己程序代码的拼字错误，却仍然检查不出问题所在。所以，有些初学者常因为自己粗心的拼写错误，再加上本身的英文能力一般，而在学习程序设计上吃尽了苦头！

2. 熟悉 HTML4 与 HTML5 语言

HTML4 和 HTML5 在语言结构上只有细微的调整，但是在元素语句上，HTML5 则增加了不少！无论是语言结构上的调整，还是元素语句上的增加，都得靠各个厂商对自家的浏览器进行改版或者升级，才能支持这些新增的部分。

Chrome、Firefox、Opera、Safari 浏览器的默认设置都会在特定的时间点上提示用户进行更新。唯独 IE 浏览器有一些麻烦的问题，对于低于 IE 9 的版本，IE 默认都不会自动更新；只有 IE 10 开始的版本，默认才会进行自动更新。

浏览器本身会提示用户进行更新，只有更新之后才有能力持续支持 HTML、CSS 与 JavaScript 的新语句。

3. CSS 搭配 HTML 的切版方式

早期的网页排版方式主要是通过表格元素来实现的，近年来的网页排版方式则主要通过 CSS 搭配 HTML 的切版方式来实现。这样的做法有明显的优点，可使得特定网页易于删减、扩充与调整版面，而主流浏览器的各个厂商也花费了不少心力支持这一项技术。

4. 变换调整网页中英文字体的方法

到 2015 年为止,读者可以观察到许多知名公司的网站中各个网页上中文采用的字体还是以新细明体居多,而标准楷体、微软雅黑体或其他字体的网页显得非常少。但是,各个网页英文字体上的表现就显得种类繁多了!

这主要是因为各平台的操作系统所内置的相同英文字体繁多,而相同中文字体非常少的缘故。为了弥补这方面的不足,网页设计团队可先行上传各主流浏览器所共同支持的特定中文字体文件到网页文件所存放的服务器中。接着再通过 CSS3 相关语句,让特定网页链接到相关的中文字体文件,这样即可让网页内容呈现出不同风格的中文字体。当然,浏览器在显示网页内容的速度上会因而变得稍微慢一点,这是网页设计团队必须权衡的问题。

5. CSS2 与 CSS3 样式应用相关的语句

CSS2 与 CSS3 在语言结构上大致相同,但是在规则语句上,CSS3 则增加了一些,而且在语言结构上稍微不同。各主流浏览器在前几年就开始相继支持 CSS3 中大部分的规则语句,设计团队可以安心在各网页的元素中应用 CSS3 的动画特效和版型外观相关的规则语句。

6. JavaScript 版型设置相关的语句

只要是 CSS 规则语句可表达的版型与外观,使用 JavaScript 一样可以办到。通过 JavaScript 来动态设置网页的版型与外观属于相当高级的方式,而且具有新增、删除、修改的高度弹性。

JavaScript 不仅可以动态修改网页内各个元素的外观,还可以整体修改各个 CSS 规则程序代码。现如今各主流浏览器都已支持这项技术,因而设计团队可自由发挥的空间更加宽广。

7. RWD Framework 的运用

虽然设计团队为实现具有 RWD 机制的网页不一定非得使用厂商们所提供的 RWD Framework,但是使用 RWD Framework 可以让设计团队节省开发**底层程序代码**所花费的大量时间。RWD Framework 本身提供了较为完整的底层程序代码,设计团队因此可以以 RWD Framework 为基础快速地在其上层进行网页的设计。

8. RWD 测试工具的操作

阶段性设计出具有 RWD 机制的网页半成品后,设计团队还应该充分利用相关的 RWD 测试工具来检测在不同屏幕尺寸下这些网页半成品是否显示正常,并持续修正,直到无瑕疵为止。

其实,各主流浏览器都已经内置了较为直觉和易于上手的 RWD 测试工具,它们的功能大同小异。

1.5 RWD 实现的机制

一般而言，同样的信息内容可被规划成 4 种不同尺寸的网页：台式机、笔记本电脑、平板电脑与智能手机。在这种情况下，早期方式会产生 4 个独立的网页文件。假如将台式机和笔记本电脑屏幕显示视为同一种网页版本，则会产生 3 个网页文件。

如今的方式则可通过 CSS3 媒体查询语句，配合 JavaScript 相关语句，融合上述各个独立的网页文件成为统一的单个网页文件。

这样的网页内容在不同尺寸的设备屏幕上显示时大致会呈现如下方式。

- 网页各元素的尺寸被缩放。
- 网页各元素内的字体大小被缩放。
- 网页内的信息内容在大尺寸屏幕中全部被显示出来，而在小尺寸屏幕里则部分被隐藏起来，通过用户的触控，再把被隐藏起来的信息内容显示于画面当中。

1.6 浏览器的支持度

截至 2015 年 8 月，全球最流行的 5 种主流浏览器按照其使用率的排名（见表 1-1）为 Chrome、Firefox、Internet Explorer (IE)、Safari、Opera。

表 1-1 主流浏览器的使用率排名

统计年月	Chrome	IE	Firefox	Safari	Opera
2015 年 08 月	64.0 %	6.6 %	21.2 %	4.5 %	2.2 %
2014 年 12 月	61.6 %	8.0 %	23.6 %	3.7 %	1.6 %
2013 年 12 月	55.8 %	9.0 %	26.8 %	3.8 %	1.9 %
2012 年 12 月	46.9 %	14.7 %	31.1 %	4.2 %	2.1 %
2011 年 12 月	34.6 %	20.2 %	37.7 %	4.2 %	2.5 %
2010 年 12 月	22.4 %	27.5 %	43.5 %	3.8 %	2.2 %
2009 年 12 月	9.8 %	37.2 %	46.4 %	3.6 %	2.3 %
2008 年 12 月	3.6 %	46.0 %	44.4 %	2.7 %	2.4 %

上述 5 种浏览器，除了 IE 浏览器之外，默认都会进行自动更新，而且都支持 JavaScript 语言与 CSS3 媒体查询语句。IE 浏览器从第 9 版开始才支持 CSS3 媒体查询语句。另外，新版的 Safari 浏览器目前只运行在苹果公司所开发的操作系统上。所以，读者如果要测试 RWD 各阶段的成果，建议使用 Chrome、Firefox 或 Opera 浏览器。下面列出上述 3 种浏览器的下载网址。

- Chrome: <http://www.google.com/chrome>。该浏览器的显示画面如图 1-5 所示。

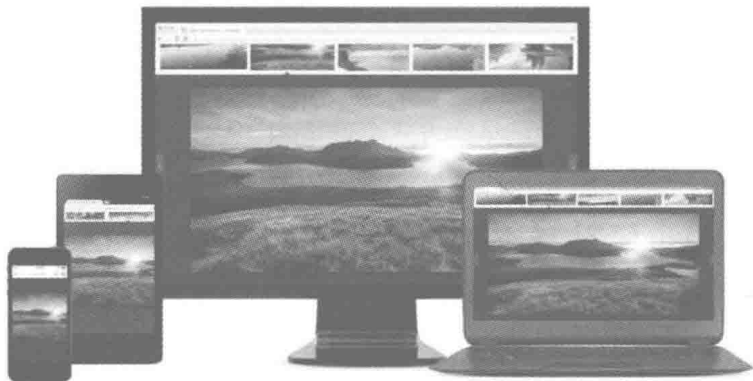


图 1-5 Chrome 浏览器在各个设备不同屏幕上显示的样子

- Firefox: <http://mozilla.com.tw/firefox>。该浏览器的显示画面如图 1-6 所示。

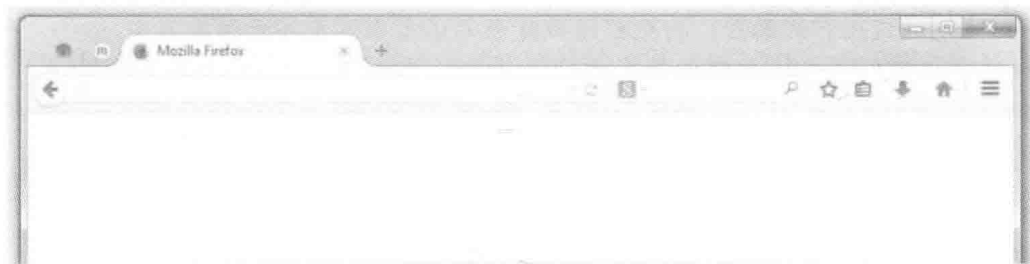


图 1-6 Firefox 浏览器的外观

- Opera: <http://www.opera.com/zh-tw>。该浏览器的显示画面如图 1-7 所示。

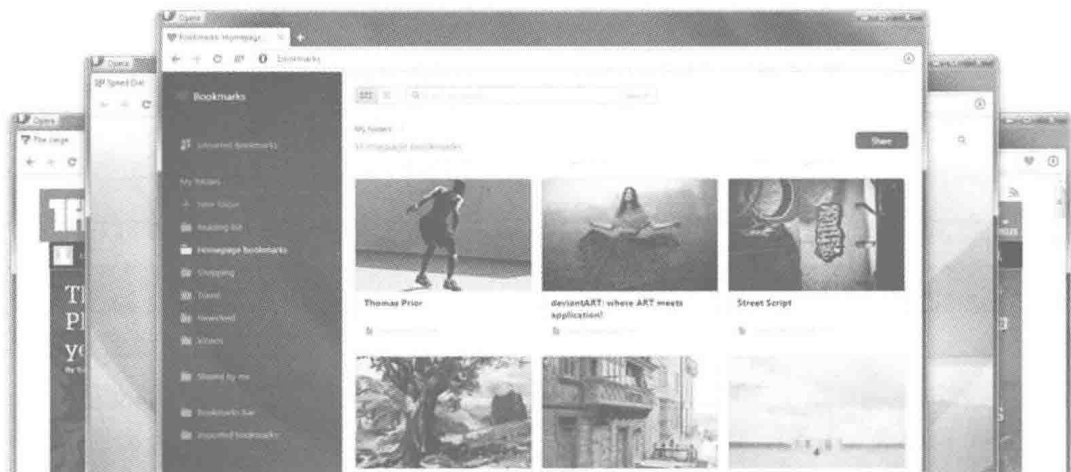


图 1-7 Opera 浏览器的外观

第 2 章 网页的新旧切版方式和字体资源

2.1 程序代码的编辑器

读者在练习本书各章节的范例时，或许已经习惯了使用某一种商业或免费的网页程序代码编辑器。而对于尚未使用这类编辑器的读者，请参考笔者推荐的两款免费编辑器，择一使用即可。

- Notepad++: <http://notepad-plus-plus.org>。
- Sublime Text: <http://www.sublimetext.com>。

其操作方式，笔者在此就不再赘述了，相信大家已经具备较为熟练的编辑器操作技能了！

2.2 CSS 搭配 HTML4 切版方式

HTML4 时代的切割版面（切版）方式被持续沿用至今，就是通过 HTML 的 div 元素语句和 CSS 相关语句来实现的切版功能。

2.2.1 开始切版前的 HTML 基本程序代码结构

为了让读者用于练习的程序代码显得更为简明扼要，笔者一开始就使用 HTML5 的基本程序代码结构作为切版的基础。

<pre><!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <title>网页的标题</title> <style> 切版程序代码关于 CSS 的部分... </style> </head> <body> 切版程序代码关于 HTML 的部分... </body> </html></pre>	让浏览器知道此为 HTML 文件。 html 元素的开始。 head 元素的开始。 解读此文件内容的字符集（编码方式）为 UTF-8 万国码格式。 设置此网页的标题。 style 元素的开始。 style 元素的结束。 head 元素的结束。 body 元素的开始。 body 元素的结束。 html 元素的结束。
--	--

2.2.2 规划要创建的版面

设计团队成员要规划哪一种网页的版面，询问过项目负责人员之后，即可开始规划出目标版面。在此，笔者先以相当简单的版面为例，让读者学习如何切版。版面范例如图 2-1 所示。

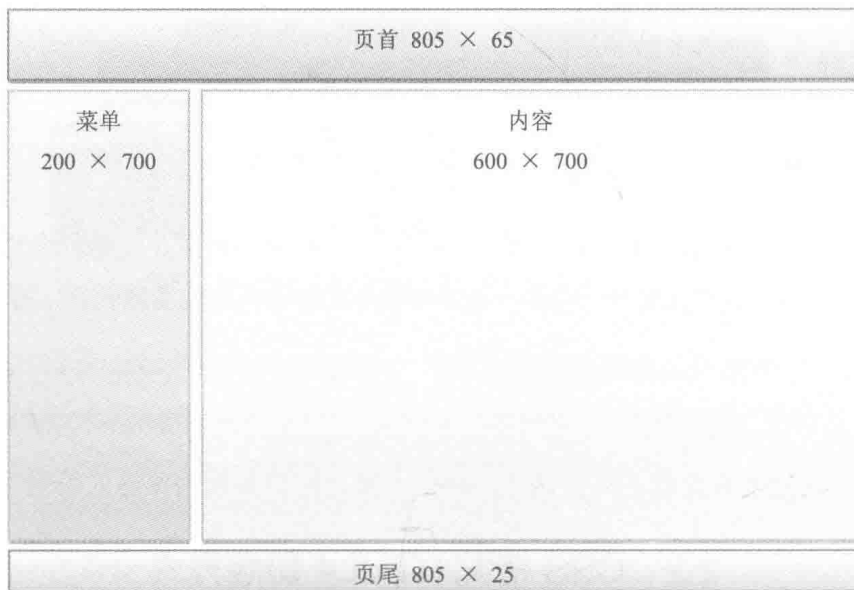


图 2-1 版面范例

在上述版面中，可以看到 4 个区块：页首、菜单、内容与页尾。每个区块间留有大致 5 个像素的间距，每个区块都被笔者标上以像素为单位的“宽度×高度”尺寸。

首先读者可在 HTML 文件中，**按照从上到下、从左到右的顺序**，提示自己要设置的区块个数与名称。

<pre>... <body> 页首 菜单 内容 页尾 </body> ...</pre>	先把提示文字按照顺序写在 body 元素内。
---	------------------------

在浏览器窗口中，本阶段程序代码的输出结果如图 2-2 所示。

页首 菜单 内容 页尾

图 2-2 版面输出范例 1

虽然在程序代码当中，我们是纵向编写 body 元素的内文，但是浏览器默认以横向文字来显示。