

# CSS世界

张鑫旭 © 著



**张鑫旭经验沉淀**  
十年实践精华系统提炼

茫茫星海中鲸鱼永不犯错的航行，  
那是因为有星辰的指引；  
芸芸CSS世界中想要不断突破瓶颈，  
需要有本书的指引。

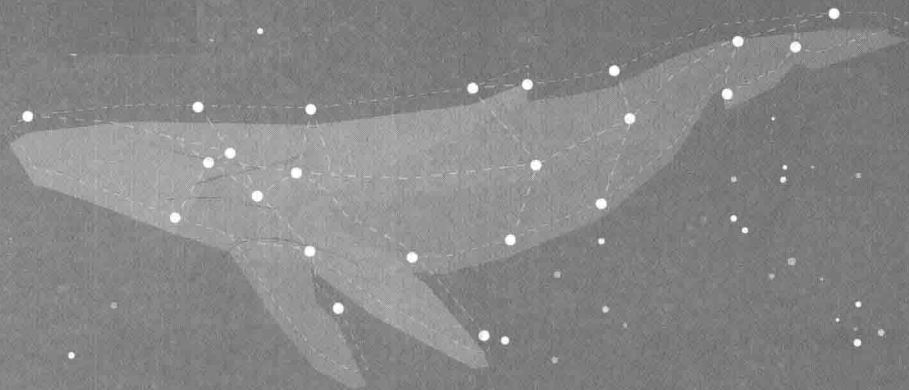
颠覆对CSS的认知  
理论与实践相得益彰



中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS



# CSS世界

张鑫旭 © 著

人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

CSS世界 / 张鑫旭著. -- 北京 : 人民邮电出版社,  
2017.12  
ISBN 978-7-115-47066-9

I. ①C… II. ①张… III. ①网页制作工具 IV.  
①TP393.092.2

中国版本图书馆CIP数据核字(2017)第265096号

## 内 容 提 要

本书从前端开发人员的需求出发,以“流”为线索,从结构、内容到美化装饰等方面,全面且深入地讲解前端开发人员必须了解和掌握的大量的CSS知识点。同时,作者结合多年的从业经验,通过大量的实战案例,详尽解析CSS的相关知识 with 常见问题。作者还为本书开发了专门的配套网站,进行实例展示、问题答疑。

作为一本CSS深度学习的书,书中介绍大量许多前端开发人员都不知道的CSS知识点。本书语言通俗易懂,内容深入浅出,并结合实战经验,更适合对CSS有所了解的前端开发人员阅读。通过阅读本书,读者会对CSS世界的深度和广度有一个全新的认识。

- 
- ◆ 著 张鑫旭  
责任编辑 杨海玲  
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>  
北京鑫正大印刷有限公司印刷
  - ◆ 开本: 800×1000 1/16  
印张: 21.5  
字数: 488千字 2017年12月第1版  
印数: 1-2400册 2017年12月北京第1次印刷

---

定价: 69.00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

## 我为什么会写这本书

我是一个利他主义非常明显的人，这多半与小时候都是在他人的帮助下成长有关，帮助我的有亲戚、邻居、老师，以及很多不认识的人。所以，现在的自己总是很乐意帮助他人成长。

我从 2007 年读大三的时候开始接触并使用 CSS，到现在已经整整 10 年了，在这 10 年时间里，我从未间断过对 CSS 的研究和学习。现在想想，能够坚持下来还真是不容易，其核心动力其实就是上面的“帮助他人，成就自我”。

开始的时候，我和大多数人一样，因为 CSS 简单，一开始成长很快，页面写多了之后还能够总结出一些准则之类的东西，并自我感觉良好，或许是自己运气好，误打误撞走出了庐山幻境，突破了学习 CSS 的一个又一个瓶颈。但是，在与诸多同行的邮件交流中我发现，很多 CSS 开发人员感到迷茫，职位得不到重视，技术也无法提高，我感觉邮件的交流一次最多只能帮助一个人，效率实在太低。

人在做抉择的时候是需要有一些指引的。实际上，很多年前，我自己曾犹豫过，是否要继续深入学习 CSS，探索每一个边界，因为对于个人而言，这会是一件吃力不讨好的事情，对于 CSS 这门语言，3 年学习 80 分和 10 年学习 90 分对于产品价值的区别其实有限。但那一封封交流邮件坚定了自己的方向，艰苦的路让我一个人走就好了，等我踏遍整个 CSS 世界，再把完整的地图绘制出来，岂不就能帮助更多人了？

所以，随着自己的不断前行，身边的人越来越少，少到好像就我一个人，无比孤寂的时候，让我坚持下来的就是“日后可以帮助更多人，是很有价值的”的信念。

10 年过去了，10 年的努力和付出终于开始开花结果，而其中一个果实就是《CSS 世界》这本书。

10 年风雨积累，踏遍 CSS 世界的千山万水，哪里有美景，哪里有秘境，哪里是陷阱，哪里是路径，我全了然于心。我这样做为的就是给予清晰明确的指引，拓展对 CSS 世界的认知，挖掘 CSS 的潜力，帮助他人突破一个又一个 CSS 学习的瓶颈。

## 为何需要这本 CSS 进阶书

大家应该都注意到了，最近行业非常缺前端开发人员，前端开发人员培训机构也如雨后春

算般大量涌现。拨开眼前的面纱，定睛一看，会发现，缺的其实不是前端，而是优秀的、有资历的、技术有深度的前端开发人员。

通过和一些前端同行、某些人力资源接触和我收到的诸多简历我发现，目前的现状是这样的：行业里有很大一拨儿人，也自称为前端开发人员，但他们仅仅是可以根据设计稿写出页面这种水平。换句话说，就是会 HTML 和 CSS 以及一点儿 JavaScript。环顾四周，这种程度的人实在是太多了，完全没有任何技术上的优势。虽然这些也是前端开发人员，但是公司要抢的前端开发人员并不是这类人。

为什么会这样呢？

因为 CSS 这门语言入门实在是太简单了，比如说我夫人，完全不懂代码，我手把手教她 1 个星期，写出一个长得像某某网首页的页面绝对是没问题的，因为 CSS 常用属性就那么多，且鲜有逻辑，无须算法，熟记各个属性值对应的特性就能上手了。所以，很多没有编程基础的人，就通过 HTML 和 CSS 进入了这个圈子。但当他们发现自己可以很愉快地实现页面的时候，就会觉得 CSS 也就这样，导致困于庐山，止步不前，就算日后听到或见到“CSS 深入很难”的言论并打算着手精进，也不知道接下来该怎么走、如何突破现有的瓶颈，于是就产生了迷茫。

这类高不成低不就的前端人员急需本书深入“CSS 世界”，突破瓶颈，告别迷茫。

在这个世界上，越是看似简单的东西反而越是难以深入，就好比为何  $1+1$  等于 2？CSS 这门语言也是如此。很多自认为学了 CSS 有八九成的人，实际上仅仅是熟记 CSS 手册中的各种属性，或者理解一些 CSS 概念，再进一步，甚至对某一两个 CSS 属性有过深入的分析。但是，这些人依然无法理解很多页面上看似简单的现象（我想更多的是根本就没在意这些现象），也无法基于现有的规则创造一些完全创新的 CSS 实现，仅仅停留在熟练地使用这种程度。

为什么会这样呢？那是因为 CSS 是一门有别于传统程序语言的语言。绝大多数编程语言，比方说 JavaScript，各种字符串、数组、方法，记住一个就是一个，使用的时候，`forEach()` 就是循环，`replace()` 就是替换，不用担心执行 `replace()` 的时候字符串突然增加了！很多人就是用这种思路学习 CSS 的，导致很快就遇到了天花板。为什么呢？这是因为，在 CSS 的世界里，页面上的任何看似简单的呈现都是由许许多多 CSS 属性共同作用的结果。例如，对于一个图片浮动，单纯认为只有 `float` 在工作是不全面的，实际上，行高、字体、鼠标手形等都在暗地里“搞鬼”，此时如果仅仅套用一两个 CSS 属性值应有的表现来理解，是根本理解不了的。换句话说，有些人的 CSS 水平之所以停滞不前，是因为他们没有把所有的 CSS 当作一个整体，放在一个完整的世界中去看待。所以，没有比“学 CSS 看看 CSS 中文手册就够了”更愚蠢的言论了，手册仅仅是表层的、独立的一些特性，每个 CSS 属性在 CSS 世界中都是有其存在的原因的，都是和其他多个 CSS 属性发生着千丝万缕的关系的，这背后的种种远比他们想象得要庞大很多。

而本书完完全全区别于传统教条式的手册或参考书或教程之类，一步一步带领读者接触真正丰富多彩、妙趣横生的 CSS 世界，一番畅快自在的 CSS 世界之旅下来，读者一定会得到不一样的“洗礼”，困扰多年的 CSS 成长瓶颈说不定就会不知不觉地突破了。

## 如何正确认识本书

首先，大家务必明确这一点，那就是本书的所有内容都是我个人的理解。没错，是人的理解，不是干巴巴的文档。这些理解是我自己多年持之以恒对 CSS 研究和思考后，经过个人情感润饰和认知提炼加工的产物，完全是我自己所理解和认识的 CSS 世界，可能是不具有权威性的。写这本书主要是希望通过分享自己的心得给对 CSS 感兴趣的各个领域的人以启迪，引发其思考，或者使其有不一样的感悟。

从另一方面讲，本书正是因为其内容都是经过人这个个体的加工，并融入了情感化的思维，才能做到有的放矢、通俗易懂。我们大多数人都是感性的，看伤感电影会哭，看综艺节目会笑。所以，与干巴巴的教条式的技术书籍相比，本书的表达方式和语言风格更能给人以心灵的启迪。读完之后，读者可能会有这样的想法：要是所有的技术书都这么写就好了。

另外，本书不是技术文档，也不是参考手册，因此知识点不会面面俱到，也就是说，不少 CSS 相关的内容我会忽略，例如，选择器这一部分最多提一下。同时，为了保证书的内容足够简练，简单的 CSS 语法和常规的使用在本书中基本上不会提及，只会重点分析在其他地方看不到的内容。

## 配套网站

我专门为本书制作了一个网站 (<http://www.cssworld.cn>)，在这个配套网站读者可以了解更多关于本书或者我个人的一些信息。

最后，由于很多内容都是个人理解，难免会有不准确或者让大家产生怀疑的地方，欢迎去官方论坛 <http://bbs.cssworld.cn>/对应版块进行提问或反馈。

## 特别感谢

衷心感谢人民邮电出版社的每一个人。

感谢人民邮电出版社的编辑杨海玲，她的专业建议对我帮助很大，她对细节的关注令人印象深刻，她使我的工作变得更加轻松。

感谢那些致力于提高整个行业 CSS 水平而默默努力的优秀人士，感谢那些在我成长路上指出错误的前端同仁，是你们让我在探索边界的道路上可以走得更快、更踏实。

感谢读者，你们的支持给了我工作的动力。

最后，最最感谢我的妻子丹丹，没有她在背后的爱和支持，本书一定不会完成得这么顺利。

# 目 录

第 1 章 概述	1
1.1 CSS 世界的“世界观”	1
1.2 世界都是创造出来的	3
1.3 CSS 完胜 SVG 的武器——流	4
1.3.1 何为“流”	5
1.3.2 流是如何影响整个 CSS 世界的	6
1.3.3 什么是流体布局	6
1.4 CSS 世界的开启从 IE8 开始	6
1.5 table 自己的世界	7
1.6 CSS 新世界——CSS3	7
第 2 章 需提前了解的术语和概念	8
2.1 务必了解的 CSS 世界的专业术语	8
2.2 了解 CSS 世界中的“未定义行为”	11
第 3 章 流、元素与基本尺寸	13
3.1 块级元素	13
3.1.1 为什么 list-item 元素会出现项目符号	15
3.1.2 display:inline-table 的盒子是怎样组成的	16
3.1.3 width/height 作用在哪个盒子上	16
3.2 width/height 作用的具体细节	16
3.2.1 深藏不露的 width:auto	17
3.2.2 width 值作用的细节	24
3.2.3 CSS 流体布局下的宽度分离原则	27
3.2.4 改变 width/height 作用细节的 box-sizing	29
3.2.5 相对简单而单纯的 height:auto	33
3.2.6 关于 height:100%	33
3.3 CSS min-width/max-width 和 min-height/max-height 二三事	37
3.3.1 为流体而生的 min-width/max-width	38



3.3.2	与众不同的初始值	38
3.3.3	超越!important, 超越最大	39
3.3.4	任意高度元素的展开收起动画技术	40
3.4	内联元素	42
3.4.1	哪些元素是内联元素	42
3.4.2	内联世界深入的基础——内联盒模型	42
3.4.3	幽灵空白节点	44
第4章	盒尺寸四大家族	45
4.1	深入理解 content	45
4.1.1	content 与替换元素	45
4.1.2	content 内容生成技术	57
4.2	温和的 padding 属性	73
4.2.1	padding 与元素的尺寸	73
4.2.2	padding 的百分比值	77
4.2.3	标签元素内置的 padding	79
4.2.4	padding 与图形绘制	80
4.3	激进的 margin 属性	81
4.3.1	margin 与元素尺寸以及相关布局	82
4.3.2	margin 的百分比值	87
4.3.3	正确看待 CSS 世界里的 margin 合并	87
4.3.4	深入理解 CSS 中的 margin:auto	94
4.3.5	margin 无效情形解析	97
4.4	功勋卓越的 border 属性	100
4.4.1	为什么 border-width 不支持百分比值	100
4.4.2	了解各种 border-style 类型	101
4.4.3	border-color 和 color	105
4.4.4	border 与透明边框技巧	106
4.4.5	border 与图形构建	108
4.4.6	border 等高布局技术	109
第5章	内联元素与流	111
5.1	字母 x——CSS 世界中隐匿的举足轻重的角色	111
5.1.1	字母 x 与 CSS 世界的基线	111
5.1.2	字母 x 与 CSS 中的 x-height	112
5.1.3	字母 x 与 CSS 中的 ex	113

5.2 内联元素的基石 line-height	114
5.2.1 内联元素的高度之本——line-height	114
5.2.2 为什么 line-height 可以让内联元素“垂直居中”	119
5.2.3 深入 line-height 的各类属性值	121
5.2.4 内联元素 line-height 的“大值特性”	124
5.3 line-height 的好朋友 vertical-align	126
5.3.1 vertical-align 家族基本认识	127
5.3.2 vertical-align 作用的前提	129
5.3.3 vertical-align 和 line-height 之间的关系	131
5.3.4 深入理解 vertical-align 线性类属性值	135
5.3.5 深入理解 vertical-align 文本类属性值	141
5.3.6 简单了解 vertical-align 上标下标类属性值	142
5.3.7 无处不在的 vertical-align	143
5.3.8 基于 vertical-align 属性的水平垂直居中弹框	144
第 6 章 流的破坏与保护	147
6.1 魔鬼属性 float	147
6.1.1 float 的本质与特性	147
6.1.2 float 的作用机制	151
6.1.3 float 更深入的作用机制	154
6.1.4 float 与流体布局	155
6.2 float 的天然克星 clear	157
6.2.1 什么是 clear 属性	157
6.2.2 成事不足败事有余的 clear	158
6.3 CSS 世界的结界——BFC	160
6.3.1 BFC 的定义	160
6.3.2 BFC 与流体布局	160
6.4 最佳结界 overflow	164
6.4.1 overflow 剪裁界线 border box	165
6.4.2 了解 overflow-x 和 overflow-y	166
6.4.3 overflow 与滚动条	166
6.4.4 依赖 overflow 的样式表现	169
6.4.5 overflow 与锚点定位	170
6.5 float 的兄弟 position:absolute	177
6.5.1 absolute 的包含块	178
6.5.2 具有相对特性的无依赖 absolute 绝对定位	184

6.5.3 absolute 与 text-align	191
6.6 absolute 与 overflow	193
6.7 absolute 与 clip	195
6.7.1 重新认识的 clip 属性	196
6.7.2 深入了解 clip 的渲染	198
6.8 absolute 的流体特性	199
6.8.1 当 absolute 遇到 left/top/right/bottom 属性	199
6.8.2 absolute 的流体特性	200
6.8.3 absolute 的 margin:auto 居中	202
6.9 position:relative 才是大哥	202
6.9.1 relative 对 absolute 的限制	203
6.9.2 relative 与定位	203
6.9.3 relative 的最小化影响原则	206
6.10 强悍的 position:fixed 固定定位	207
6.10.1 position:fixed 不一样的“包含块”	207
6.10.2 position:fixed 的 absolute 模拟	208
6.10.3 position:fixed 与背景锁定	209
<b>第 7 章 CSS 世界的层叠规则</b>	<b>211</b>
7.1 z-index 只是 CSS 层叠规则中的一叶小舟	211
7.2 理解 CSS 世界的层叠上下文和层叠水平	212
7.2.1 什么是层叠上下文	212
7.2.2 什么是层叠水平	212
7.3 理解元素的层叠顺序	212
7.4 务必牢记的层叠准则	214
7.5 深入了解层叠上下文	214
7.5.1 层叠上下文的特性	214
7.5.2 层叠上下文的创建	214
7.5.3 层叠上下文与层叠顺序	217
7.6 z-index 负值深入理解	219
7.7 z-index “不犯二”准则	223
<b>第 8 章 强大的文本处理能力</b>	<b>225</b>
8.1 line-height 的另外一个朋友 font-size	225
8.1.1 font-size 和 vertical-align 的隐秘故事	225
8.1.2 理解 font-size 与 ex、em 和 rem 的关系	227

8.1.3 理解 font-size 的关键字属性值	229
8.1.4 font-size:0 与文本的隐藏	231
8.2 字体属性家族的大家长 font-family	232
8.2.1 了解衬线字体和无衬线字体	233
8.2.2 等宽字体的实践价值	234
8.2.3 中文字体和英文名称	236
8.2.4 一些补充说明	237
8.3 字体家族其他成员	238
8.3.1 貌似粗犷、实则精细无比的 font-weight	238
8.3.2 具有近似姐妹花属性值的 font-style	241
8.3.3 不适合国情的 font-variant	242
8.4 font 属性	242
8.4.1 作为缩写的 font 属性	242
8.4.2 使用关键字值的 font 属性	243
8.4.3 font 关键字属性值的应用价值	246
8.5 真正了解@font face 规则	247
8.5.1 @font face 的本质是变量	247
8.5.2 @font face 与字体图标技术	255
8.6 文本的控制	258
8.6.1 text-indent 与内联元素缩进	258
8.6.2 letter-spacing 与字符间距	261
8.6.3 word-spacing 与单词间距	263
8.6.4 了解 word-break 和 word-wrap 的区别	264
8.6.5 white-space 与换行和空格的控制	265
8.6.6 text-align 与元素对齐	267
8.6.7 如何解决 text-decoration 下划线和文本重叠的问题	271
8.6.8 一本万利的 text-transform 字符大小写	273
8.7 了解:first-letter/:first-line 伪元素	274
8.7.1 深入:first-letter 伪元素及其实例	274
8.7.2 故事相对较少的:first-line 伪元素	277
第9章 元素的装饰与美化	280
9.1 CSS 世界的 color 很单调	280
9.1.1 少得可怜的颜色关键字	280
9.1.2 不支持的 transparent 关键字	282
9.1.3 不支持的 currentColor 变量	282

---

9.1.4	不支持的 rgba 颜色和 hsla 颜色	282
9.1.5	支持却鸡肋的系统颜色	283
9.2	CSS 世界的 background 很单调	285
9.2.1	隐藏元素的 background-image 到底加不加载	285
9.2.2	与众不同的 background-position 百分比计算方式	286
9.2.3	background-repeat 与渲染性能	287
9.2.4	外强中干的 background-attachment:fixed	288
9.2.5	background-color 背景色永远是最低的	289
9.2.6	利用多背景的属性 hack 小技巧	290
9.2.7	渐变背景和 rgba 背景色的兼容处理	290
第 10 章	元素的显示与隐藏	292
10.1	display 与元素的显隐	294
10.2	visibility 与元素的显隐	296
10.2.1	不仅仅是保留空间这么简单	296
10.2.2	了解 visibility:collapse	301
第 11 章	用户界面样式	302
11.1	和 border 形似的 outline 属性	302
11.1.1	万万不可在全局设置 outline:0 none	302
11.1.2	真正的不占据空间的 outline 及其应用	304
11.2	光标属性 cursor	307
11.2.1	琳琅满目的 cursor 属性值	307
11.2.2	自定义光标	314
第 12 章	流向的改变	315
12.1	改变水平流向的 direction	315
12.1.1	direction 简介	315
12.1.2	direction 的黄金搭档 unicode-bidi	318
12.2	改变 CSS 世界纵横规则的 writing-mode	320
12.2.1	writing-mode 原本的作用	321
12.2.2	writing-mode 不经意改变了哪些规则	324
12.2.3	writing-mode 和 direction 的关系	328

# 第 1 章

## 概述

要深入理解一个事物之前，最好先对其整体有个大概了解，这样才不至于蠢测管窥。如果把 CSS 比作一座大山，则我们对整体的认知就好比知道这座山的位置、山势、道路状况等，这样，当我们深入其中的时候，就不容易迷路，不会做出错误的决策。

例如，对 CSS 这门语言特性的描述就有助于对 CSS 的整体认知。具体表现为：擅长 C++ 或者 Java 之类的程序员学习 CSS 往往没有如鱼得水的感觉，其背后的原因是，典型的计算机开发语言看重逻辑思维和抽象能力，但是 CSS 这门语言本身并无逻辑可言，看重的是特性间的相互联系和具象能力。

具象往往以情感为纽带，无意识不自觉产生，是非常感性的一种能力，这往往是偏理性的程序员所不擅长的。在某些程序员眼中，CSS 属性就是干巴巴的属性，无法建立类似“人与人关系”这种很情感化的联系，于是学习 CSS 总是只得其形、不得其髓。

当然，认知可以从多个角度进行。例如，接下来要介绍的 CSS“世界观”以及 CSS 的历史故事，可以让我们多种角度同时进行认知，对 CSS 这门语言的理解更为准确和丰满。

### 1.1 CSS 世界的“世界观”

对于 CSS 这门语言，我学习和研究已经有 10 年之久，在点点滴滴的积累中，逐渐形成了一套完整的体系。在 CSS 这个世界中，CSS 并不是一个机械枯燥的语言，所有属性都是有血有肉、有着不同个性和身世的个体。不同的个体可以碰撞出不同的火花，激荡出异彩纷呈的故事。

这里，我们不妨“脑洞大开”一下：如果把 CSS 世界拍成动漫的话，会是什么样子？

首先，动漫名可以叫作《建筑神域》，讲述一群建筑魔法师为国家存亡惊心动魄战斗的故事。然后，出现了“Chrome 王国”的几位建筑魔法师日常训练的画面。只见名为 width 的魔法师手持名叫选择器的法器，准确指向称为<div>的最普通的块状建筑魔法石，口中念

道：“层叠天星，幻化有形，50%，变！”只听见一声清脆的“啪”，`<div>`魔法石宽度变成了原来的一半。然而，`width`却锁眉摇头，口中喃喃念道：“1毫秒，还不够快，还要再练，不然在和‘IE王国’的战斗中很难占得先机！”（如图1-1所示）。



图 1-1 CSS 世界观示意 1

此时，`width`突然发现前面1米之处有一块`<span>`之石，具有`class="test"`的特殊标记，立即拿出法器，念道：“类名之石，`test`为名，为我选择，出！”话音刚落，`<span>`之石蓝色荧光一闪，明眼人都能看出来，`width`魔法师和`<span>`魔法石现在处于契约状态。`width`继续念道：“层叠天星，幻化有形，50%，变！”但`<span>`魔法石却没有任何变化，此时`width`一拍自己的脑门，似乎明白了什么，转过头对旁边的`display`魔法师大声叫道：“小D，这边这边，过来帮个忙……来呀，快点……”

只见`display`迅速结束自己的练习，屁颠屁颠跑过来：“咋啦？”

“此为内联之石，我无法驾驭，你帮我重塑一下。”

“小问题！正好，魔法师技能委员会刚通过了我的一个新法术，我给你秀一秀？”

“哟，不错啊，快让我瞅瞅！”

“好嘞！”只见 display 拿出自己的法器，念道：“类名之石，test 为名，为我选择，出！”紧接着，“层叠天星，幻化有形，flex，变！”

只听见一声清脆的“啪”，在两人的合作之下，<span>魔法石宽度也变化了（如图 1-2 所示）。

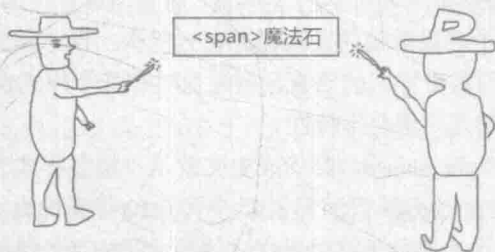


图 1-2 CSS 世界观示意 2

“哟，很酷嘛！”width 对 display 竖着大拇指称赞道。

只见 display 腼腆一笑，小手在面前轻轻一挥：“就算你这么夸我，人家也不会开心的啦……”

从上面的描述可以看出，在 CSS 世界中，HTML 是魔法石，选择器就是选择法器，CSS 属性就是魔法师，CSS 各种属性值就是魔法师的魔法技能，浏览器就是他们所在的“王国”，“王国”会不断更新法律法规（版本升级），决定是否允许使用新的魔法石（HTML5 新标签新属性），是否允许新的魔法师入“国籍”（CSS3 新属性），或者允许魔法师使用某些新技能（CSS 新的属性值），以及是否舍弃某些魔法技能（如 `display:run-in`）；操作系统就是他们所在的世界，不同的操作系统代表不同的平行世界，所以，CSS 世界有这么几个比较大的平行世界，即 Windows 世界、OS X 世界以及移动端的 iOS 世界和 Android 世界。不同世界的浏览器王国的命运不一样，例如，在 OS X 世界中，IE 王国是不存在的，而 Safari 王国却异常强大，但在 Windows 世界中，Safari 王国却异常落寞。

以上这一切就构成了完整的 CSS 世界的“世界观”。

下面回答一个很重要的问题：为何要这样认识 CSS 世界呢？

首先，将抽象的 CSS 直接和具体的现实世界相对应，更加易于理解。试想一下，对于普通人，理解魔法师和魔法石是不是要比理解 CSS 代码容易得多？其次，以完整的体系来学习 CSS 要比单纯关注属性值理解得更加深刻，可以培养从宏观层面认识与理解 CSS 的习惯。再次，这也方便我们记忆，枯燥的代码总是过目就忘，鲜活的角色总是印象深刻。最后，这样也可以让本书散发出与众不同的气质。

## 1.2 世界都是创造出来的

世界都是创造出来的。很自然，CSS 世界也是一点一点创造出来的。这世间上的事情只要发生了，都是有原因的。CSS 世界的出现也不例外。

下面我们来看一下 CSS 世界出现的历史。虽然我知道，有些人对这些历史可能不感兴趣，



但是要想深入理解 CSS 属性的一些设计原因、表现原理还真离不开当时的历史环境。

大家可能都听说过马云 1995 年去美国，第一次接触了互联网，在这个时间点，HTML 才是第一版且诞生没几年，W3C 才刚刚成立，CSS 还没出现。那时候的互联网几乎都是文字信息，显示一张图片都是要上天的感觉。

大家可能没意识到，那个时候前端的发展和现在一样快，设计师要求越来越多，HTML 也越来越庞杂。急需要其他专门负责样式的语言，据说当时有几个样式表语言，最后是 CSS 胜出了，为什么呢？它的胜出靠的是“层叠”特性。

CSS 全称是 Cascading Style Sheets，翻译成中文就是“层叠样式表”。所谓“层叠”，顾名思义，就是样式可以层层累加，比方说页面元素都继承了 12 像素的大小，某标题就可以设置成 14 像素进行叠加。发现没？这种层叠策略对于样式的显示是相当的灵活。

于是，从 1996 年 12 月 17 日 CSS1 诞生后，CSS 在样式呈现领域可谓所向披靡，没有遇到任何竞争对手。1998 年 5 月 12 日 CSS2 发布，推行内容和表现分离，表格（table）布局开始落寞。

1998 年腾讯、新浪和网易成立，当时搜狐则成立 1 年不到。那个时候是门户的时代，人们更关注的是信息的获取，所以网站的功能主要就是信息展示，信息是什么？在那个时代，在互联网领域，信息就是图片和文字。换句话说，那时候的网站前端技术关心的是图片和文字的呈现，而 CSS2（包括 9 年之后，也就是 2007 年才出现的 CSS2.1）都是为图文展示服务的。

我再重复一遍：CSS 世界的诞生就是为图文信息展示服务的。这句话在本书中会非常频繁地出现，知道这一点你就会明白很多事情。

好，下面让我们回到本节开头的那句话——“世界都是创造出来的”！为何我要反复强调这句话呢？如果站在造物主的角度去思考 CSS 世界的种种表现，很多问题就会迎刃而解。

现在给你机会当一回造物主，让你自己重新构建一个 CSS 世界，唯一的要求就是，这个世界要非常便于图片和文字的呈现，你会去如何构建呢？

### 1.3 CSS 完胜 SVG 的武器——流

在 2003 年 1 月，SVG 1.1 被确立为 W3C 标准。你没看错，是 2003 年。要知道，CSS 2.1 是 2007 年才发布的。考虑到 SVG 开始火起来是最近几年，也就是差不多 10 年的时间，SVG 都默默无闻，鲜有人问津，到底是怎么回事呢？

很多人认为 SVG 的竞争对手是 Flash。对，是竞争对手。但是，现在看来，SVG 显然要比 Flash 优秀很多，SVG 开放、标准，和 CSS 和 JavaScript 都能很方便地进行交互，如果单纯 SVG 和 Flash 比，难说谁胜谁负。在我看来，造成 SVG 被冷落 10 年的原因不是别的，正是看似毫不相关的 CSS，SVG 是被 CSS 给打败的。

正如上面提到的，在很长一段时间里，网站的主要功能都是图片和文字信息的展示，但是，SVG 的强项是图形，其文字内容的呈现在不敢恭维。举个例子，在 CSS 中写上一段文字，这段文字会自然换行、多行显示，于是，可以像书本一样阅读；但是，在 SVG 中，文字要自动折行，感觉有点儿赶鸭子上架——强人所难。人家一看，SVG 连基本的文字排版都做不好，要 SVG