



计算机基础与实训教材系列

HTML5+CSS3网页设计

实例教程

杨福贞 何永峰 李岩 编著



(理论→实例→上机→习题)4阶段教学模式

任务驱动的讲解方式，方便学习和教学

众多典型的实例操作，注重培养动手能力

PPT电子教案及素材免费下载，专业的网上技术支持



清华大学出版社

计算机基础与实训教材系列

HTML5+CSS3网页设计

实例教程

杨福贞 何永峰 李岩 编著

清华大学出版社
北 京

内 容 简 介

本书循序渐进地讲述 Web 网页设计技术,从基本概念到 HTML5 网页设计、CSS3 样式控制以及综合实例,主要内容包括网页设计新时代、HTML5 的演进、创建 HTML5 文档、表格与列表、图片与富媒体、HTML5 表单的应用、CSS3 概述、CSS3 选择器、文本与字体、应用 CSS3 的属性、页面布局与媒介查询、变形和变换以及动画等,并且运用大量实例对各种关键技术进行深入浅出的分析。

本书内容丰富、结构合理、思路清晰、语言简练流畅、示例翔实,既适合作为高等院校计算机相关专业的教材,也可作为网页设计应用开发人员的参考书。

本书的电子课件、习题答案和实例源代码可以到 <http://www.tupwk.com.cn/downpage> 网站下载,也可通过扫描前言中的二维码下载。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

HTML5+CSS3 网页设计实例教程 / 杨福贞, 何永峰, 李岩 编著. —北京: 清华大学出版社, 2019
(计算机基础与实训教材系列)
ISBN 978-7-302-52500-4

I. ①H… II. ①杨… ②何… ③李… III. ①超文本标记语言—程序设计—教材 ②网页制作工具—教材 IV. ①TP312.8 ②TP393.092.2

中国版本图书馆 CIP 数据核字(2019)第 043133 号

责任编辑: 胡辰浩

装帧设计: 孔祥峰

责任校对: 牛艳敏

责任印制: 沈 露

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 190mm×260mm 印 张: 25.5 字 数: 652 千字

版 次: 2019 年 4 月第 1 版 印 次: 2019 年 4 月第 1 次印刷

定 价: 68.00 元

产品编号: 077832-01

前 言

Internet 正在经历惊人的巨变,不仅有浩如烟海的用户人群,还有复杂多样的联网设备,网络的连接速度也与日俱增。简单的网页设计技术已经不能满足需求,HTML5 和 CSS3 技术的更新,更加优化了网页设计的技术标准。

HTML5+CSS3 以标准布局和精美样式,实现了网页内容和样式的分离,使网页样式布局 and 美化达到空前的高度,成为 Web 标准中不可替代的技术规范。本书以知识点示例、章节综合实例和全书综合实例等形式,全面涵盖网页设计的基础知识、HTML5 和 CSS3 技术等。

本书共 13 章,从 Web 网页设计的基本概念出发,由浅入深地详细讲述了包括网页设计新时代、HTML5 的演进、创建 HTML5 文档、表格与列表、图片与富媒体、HTML5 表的应用、CSS3 概述、CSS3 选择器、文本与字体、应用 CSS3 的属性、页面布局与媒介查询、变形和变换以及动画等内容。在讲述网页设计的各种技术时,运用丰富的实例,注重培养读者解决实际问题的能力,使大家快速掌握应用 HTML5+CSS3 进行网页设计的操作技术。

本书内容丰富、结构合理、思路清晰、语言简练流畅、示例翔实。每一章的引言部分概述了该章的作用和内容。在每一章的正文中,结合所讲述的关键技术和难点,穿插了大量极富实用价值的示例。每一章末尾都安排了有针对性的思考题和练习题,思考题有助于读者巩固所学的基本概念,练习题有助于培养读者的实际动手能力,增强对基本概念的理解和实际应用能力。

本书面向高等院校计算机相关专业师生,以及网页设计爱好者和制作人员。既适合作为高等院校网页设计课程的教材,也可作为网页设计应用开发人员的参考书。

参与本书编写的人员均为高等院校从事计算机课程教学和研究的教师,全书共 13 章,其中集宁师范学院的杨福贞负责第 1、2、7、8、9 章的编写;集宁师范学院的何永峰负责第 3、4、10、12 章的编写;集宁师范学院的李岩负责第 5、6、11、13 章的编写。另外,参加本书编写的人员还有薛山、梁琦、高宇飞、赵国桦、丁鑫、贾圣杰、武红欣、张晓菊、陈光训、吴超群、郑玉祥、付君泽、黄怀春、靳廷喜等人。全书整体框架设计由杨福贞负责,组稿工作由李岩负责,统稿工作由何永峰负责,薛山及梁琦等协助编者进行材料搜集工作。

本书在编写过程中参考了国内同行和广大实践工作者的诸多文献与资料,并引用了部分材料与成果。在此一并致以深深的敬意和衷心的感谢。

尽管全体编写者都已倾尽全力而为，但由于水平有限，难免有疏漏和不足之处，希望同行专家给予指正。我们的电话是 010-62796045，信箱是 huchenhao@263.net。

本书的电子课件、习题答案和实例源代码可以到 <http://www.tupwk.com.cn/downpage> 网站下载，也可通过扫描下方二维码下载。



作者
2018年10月

目 录

第1章 网页设计新时代	1
1.1 Web网页设计原则	1
1.1.1 Web标准	1
1.1.2 Web标准与浏览器的兼容性	3
1.1.3 可访问性	3
1.1.4 结构化文档的Web	4
1.2 HTML5概述	4
1.2.1 传统HTML与XHTML	4
1.2.2 HTML5是如何形成的	5
1.2.3 HTML5的设计原则	7
1.3 HTML5页面的特征	9
1.3.1 使用HTML5结构化元素	10
1.3.2 使用CSS3美化HTML5文档	13
1.4 Web网页设计基础	14
1.4.1 网页的基本元素	14
1.4.2 网页设计常用技术	17
1.4.3 网页设计常用工具	18
1.5 编写一个简单的Web页面	20
1.5.1 搭建浏览环境	20
1.5.2 检测浏览环境是否支持	20
1.5.3 使用HTML5编写一个简单的Web页面	21
1.6 本章小结	22
1.7 思考和练习	22
第2章 HTML5的演进	24
2.1 HTML与XHTML基础	24
2.1.1 标签与元素	24
2.1.2 特性组	26
2.1.3 HTML和XHTML实例	29
2.2 认识HTML5	31
2.2.1 用XHTML 1.0标记页面	31
2.2.2 84.8%的标记可以保留	32
2.3 关于<head>	33
2.3.1 更完美的DOCTYPE	33
2.3.2 在HTML5中声明语言	34
2.3.3 字符编码	35
2.3.4 简单易记	35
2.4 从HTML与XHTML到HTML5	35
2.4.1 XHTML 1.0风格的“Hello World!”	35
2.4.2 HTML 4.01风格的“Hello World!”	36
2.4.3 HTML5“松散”风格的“Hello World!”	36
2.4.4 HTML5“严格”风格的页面	37
2.5 让HTML5得到跨浏览器支持	37
2.5.1 浏览器如何处理未知元素	37
2.5.2 shiv	39
2.5.3 IE打印保护器	40
2.5.4 声明块级元素	40
2.6 HTML5样板页面	40
2.6.1 不再有type属性	41
2.6.2 填充材料与替换	42
2.7 HTML5页面验证	42
2.7.1 HTML5验证器	42
2.7.2 更新和验证“贝克小姐”页面	44
2.8 新的结构化元素	46
2.8.1 块级元素与行内元素	47

2.8.2	结构化构建块<div>、<section>和 <article>	47	4.5	嵌套表格	102
2.8.3	标题: <header>、<footer>、 <hgroup>以及<h1>~<h6>	50	4.6	易访问表格	103
2.8.4	更多的结构化元素: <nav>、<aside>、 <figure>以及<figcaption>	50	4.6.1	如何线性化表格	104
2.8.5	将一个简单的页面转换成HTML5 页面	52	4.6.2	使用id、scope及headers特性	105
2.9	本章小结	54	4.6.3	创建易访问表格	106
2.10	思考和练习	55	4.7	使用列表	107
第3章	创建HTML5文档	56	4.7.1	使用元素创建无序列表	108
3.1	HTML5文档结构	56	4.7.2	有序列表	108
3.1.1	构建基本的文档结构	56	4.7.3	定义列表	110
3.1.2	用元数据元素说明文档	58	4.7.4	列表嵌套	111
3.1.3	使用脚本元素	61	4.8	表格应用实例	111
3.2	构建文档主体	62	4.9	本章小结	113
3.2.1	基本文本格式化	62	4.10	思考和练习	114
3.2.2	链接与导航	67	第5章	图片与富媒体	115
3.3	描述文本级语义	78	5.1	在网页中添加图片	115
3.3.1	描述文本级语义的元素	79	5.1.1	使用元素添加图片	115
3.3.2	使用文本标记的实例	84	5.1.2	使用图片作为链接	117
3.4	编辑文本	86	5.1.3	选择正确的图片格式	118
3.5	使用字符实体显示特殊字符	87	5.2	HTML5的视频	119
3.6	注释	87	5.2.1	视频格式	120
3.7	本章小结	88	5.2.2	浏览器支持	121
3.8	思考和练习	88	5.2.3	添加视频源	121
第4章	表格与列表	90	5.2.4	track元素	124
4.1	表格介绍	90	5.2.5	更多的视频设置	126
4.2	基本表格元素与特性	93	5.3	HTML5的音频	127
4.2.1	使用<table>元素创建表格	93	5.3.1	音频编解码器	128
4.2.2	创建基本表格	95	5.3.2	浏览器支持情况	128
4.3	为表格添加标题	96	5.3.3	添加音频源	129
4.4	表格区域分组	97	5.3.4	使用jPlayer	129
4.4.1	使用colspan特性跨越列	97	5.4	canvas	129
4.4.2	使用rowspan特性跨越行	98	5.4.1	基于像素的自由	129
4.4.3	将表格分解为表头、表体及表尾	99	5.4.2	添加/实现canvas	130
4.4.4	使用<colgroup>元素对列进行分组	101	5.4.3	canvas的威力与潜力	137
4.4.5	使用<col>元素在列间共享样式	102	5.5	SVG	139
			5.5.1	SVG与canvas的选择	139
			5.5.2	用SVG发布向量	140
			5.6	富媒体页面的设计实例	142
			5.7	本章小结	144

5.8 思考和练习	144	第7章 CSS3概述	183
第6章 HTML5表单的应用	145	7.1 CSS基础	183
6.1 认识表单	145	7.1.1 CSS演变历史	183
6.2 使用<form>元素创建表单	147	7.1.2 CSS的版本	184
6.2.1 action特性	147	7.1.3 CSS规则	187
6.2.2 method特性	148	7.1.4 继承	190
6.2.3 id特性	148	7.2 CSS规则的添加位置	191
6.2.4 name特性	148	7.2.1 用<link>元素链入外部样式表	192
6.2.5 enctype特性	148	7.2.2 用<style>元素导入外部样式表	193
6.2.6 accept-charset特性	148	7.2.3 使用外部样式表	193
6.2.7 novalidate特性	148	7.2.4 外部样式表的优势	196
6.2.8 target特性	149	7.3 CSS属性	197
6.2.9 autocomplete特性	149	7.4 CSS3新增特性	198
6.3 <form>元素中的表单控件	149	7.4.1 CSS3模块化方式	198
6.3.1 文本输入控件	149	7.4.2 CSS3新增属性及伪类	200
6.3.2 新的HTML5输入控件类型与特性	153	7.4.3 CSS3的优点	201
6.3.3 按钮	159	7.5 一个基本实例	202
6.3.4 复选框	161	7.6 本章小结	205
6.3.5 单选按钮	163	7.7 思考和练习	205
6.3.6 下拉选择框	164	第8章 CSS3选择器	206
6.3.7 文件选择框	167	8.1 选择器概述	206
6.3.8 隐藏控件	168	8.2 选择器	208
6.3.9 新的HTML5表单元素	168	8.2.1 通用选择器	208
6.4 创建一个联系方式表单	170	8.2.2 类型选择器	208
6.5 使用<label>元素为控件创建 标签	172	8.2.3 类选择器	208
6.6 使用<fieldset>及<legend>元素 组织表单结构	173	8.2.4 id选择器	209
6.7 焦点	174	8.2.5 子选择器	209
6.7.1 标签遍历顺序	175	8.2.6 后代选择器	209
6.7.2 快捷键	176	8.2.7 相邻兄弟选择器	210
6.8 disabled与readonly控件	176	8.2.8 一般兄弟选择器	210
6.9 向服务器发送表单数据	178	8.2.9 子选择器与相邻兄弟选择器的 用途	210
6.9.1 HTTP get方法	178	8.2.10 特性选择器	211
6.9.2 HTTP post方法	179	8.3 CSS3选择器	213
6.10 创建更有用的表单字段	179	8.3.1 组合器	213
6.11 本章小结	181	8.3.2 特性选择器和子串选择器	214
6.12 思考和练习	182	8.3.3 UI元素状态伪类	218
		8.3.4 :target伪类	220

8.3.5 结构伪类	223	9.5.4 选择正确的字宽	258
8.3.6 伪元素	229	9.5.5 设置垂直间距	258
8.3.7 否定伪类	231	9.5.6 用网格设计字体	258
8.4 浏览器支持	232	9.6 使用CSS3控制文本格式	261
8.6 应用实例	232	9.6.1 控制文本溢出	261
8.7 本章小结	235	9.6.2 从基线开始垂直对齐文本	261
8.8 思考和练习	235	9.6.3 控制单词中各个字符之间的空白	262
9.6.4 调整字间距	262	9.6.4 调整字间距	262
9.6.5 打断长单词	263	9.6.5 打断长单词	263
9.6.6 控制空白和换行符	263	9.6.6 控制空白和换行符	263
9.6.7 打印断字	264	9.6.7 打印断字	264
9.6.8 控制文本标点符号	265	9.6.8 控制文本标点符号	265
9.6.9 控制非拉丁Web字体的显示	266	9.6.9 控制非拉丁Web字体的显示	266
9.6.10 使用连字和其他OpenType字体功能	267	9.6.10 使用连字和其他OpenType字体功能	267
9.7 本章小结	268	9.7 本章小结	268
9.8 思考和练习	268	9.8 思考和练习	268
第9章 文本与字体	237	第10章 应用CSS3的属性	270
9.1 用CSS控制文本	237	10.1 颜色与透明度	270
9.1.1 字库和字体	238	10.1.1 RGB	270
9.1.2 font-family属性	238	10.1.2 RGBA透明度	271
9.1.3 font-size属性	239	10.1.3 HSLa	272
9.1.4 font-weight属性	240	10.1.4 不透明度	273
9.1.5 font-style属性	241	10.2 背景	274
9.1.6 font-variant属性	242	10.2.1 background-clip属性	274
9.2 用CSS格式化文本	243	10.2.2 background-origin属性	275
9.2.1 color属性	243	10.2.3 background-size属性	277
9.2.2 text-align属性	243	10.2.4 多重背景	279
9.2.3 vertical-align属性	244	10.3 边框	281
9.2.4 text-decoration属性	245	10.3.1 border-radius属性	282
9.2.5 text-indent属性	245	10.3.2 border-image属性	284
9.2.6 text-shadow属性	246	10.4 下拉阴影	286
9.2.7 text-transform属性	246	10.4.1 box-shadow属性	286
9.2.8 letter-spacing属性	247	10.4.2 text-shadow属性	288
9.2.9 word-spacing属性	248	10.5 渐变	289
9.2.10 white-space属性	248	10.6 检测支持和辅助其他浏览器	293
9.2.11 direction属性	249	10.6.1 使用Modernizr	294
9.2.12 文本伪类	250	10.6.2 CSS3 Pie	294
9.3 文本样式化实例	250		
9.4 使用@font-face自定义字体	252		
9.4.1 解析@font-face语法: @font-face声明	252		
9.4.2 @font-face的可靠语法	253		
9.4.3 为网页添加自定义字体的实例	253		
9.5 使用CSS3调整字体	256		
9.5.1 设置font-family	256		
9.5.2 设置字号	257		
9.5.3 选择字形的粗细	257		

10.7	组合CSS3效果实例	295	12.1.2	元素的变形效果	336
10.8	本章小结	297	12.1.3	修改变形对象	342
10.9	思考和练习	297	12.1.4	浏览器对CSS变形的支持	346
第11章	页面布局与媒体查询	299	12.2	CSS变换和CSS动画	348
11.1	盒子模型	299	12.3	CSS变换	348
11.1.1	一个演示盒子模型的示例	300	12.3.1	使用transition-property设置 可变换的属性	349
11.1.2	border属性	301	12.3.2	使用transition-duration控制变换的 持续时间	351
11.1.3	padding属性	303	12.3.3	transition-timing-function、三次 贝塞尔曲线和steps()函数	351
11.1.4	margin属性	304	12.3.4	使用transition-delay推迟变换的 起始时间	353
11.1.5	内容盒子的尺寸	305	12.3.5	多个变换值和简写形式的变换 属性	354
11.2	使用CSS定位与布局	309	12.3.6	简写变换属性时值的排序	354
11.2.1	正常流	309	12.3.7	浏览器对CSS变换的支持	354
11.2.2	position属性	310	12.4	使用CSS动画制作关键帧动画	355
11.2.3	盒子偏移属性	310	12.4.1	创建一个简单动画	356
11.2.4	相对定位	310	12.4.2	使用@keyframes控制动画	357
11.2.5	绝对定位	311	12.4.3	使用animation-timing-function 属性的计时函数	358
11.2.6	固定定位	312	12.4.4	使用animation-delay修改动画的 起始方式	359
11.2.7	z-index属性	313	12.4.5	控制动画的运行次数	359
11.2.8	使用float属性实现流动	314	12.4.6	结合animation-direction	360
11.2.9	clear属性	316	12.4.7	控制元素在动画前后的表现	360
11.3	构建页面布局实例	317	12.4.8	暂停动画	362
11.4	媒体查询	320	12.4.9	简略形式的动画属性以及用逗号 分隔的animation-*值	362
11.5	CSS3布局	322	12.4.10	浏览器对CSS动画的支持	363
11.5.1	CSS3定位布局模块	322	12.5	CSS动画+JavaScript	363
11.5.2	CSS3碎片模块	322	12.6	本章小结	364
11.5.3	多列布局模块	322	12.7	思考和练习	364
11.5.4	CSS3区域模块	325	第13章	网页设计综合实例	366
11.5.5	CSS3排除和形状模块	326	13.1	设计旅游网站网页	366
11.5.6	CSS3分页媒体模块	327	13.1.1	页面的设计	366
11.5.7	用于分页媒体模块的CSS生成 内容	328	13.1.2	全局样式的设计	368
11.5.8	弹性盒子布局模块	329			
11.5.9	CSS网格布局模块	331			
11.6	本章小结	332			
11.7	思考和练习	332			
第12章	变形、变换和动画	334			
12.1	CSS的2D和3D变形	334			
12.1.1	变形属性和变形函数	335			

13.1.3	网页首部(top区)	368
13.1.4	主内容区(main区)	371
13.1.5	页尾区(footer区)	379
13.2	设计网页游戏	380
13.2.1	游戏简介	380
13.2.2	界面布局的设计	381

13.2.3	数据模型的设计	385
13.2.4	游戏的逻辑实现	388
13.2.5	游戏的完整代码	397

参考文献	398
------	-----

第 1 章

网页设计新时代

从 2010 年开始, HTML5 和 CSS3 就一直是互联网技术中最受关注的两个话题。2010 年, 在 MIX10 大会上微软的工程师在介绍 IE9 时, 从前端技术的角度把互联网的发展分为三个阶段: 第一阶段是以 Web 1.0 为主的网络阶段, 前端主流技术是 HTML 和 CSS; 第二阶段是 Web 2.0 的 Ajax 应用阶段, 热门技术是 JavaScript/DOM/异步数据请求; 第三阶段为 HTML5+CSS3 阶段, 这两者相辅相成, 使网页设计进入一个崭新的时代。本章将重点介绍 HTML5 语言基础、HTML5 的由来及其要解决的问题、指导进行网页开发的设计原则以及 HTML5 带来的新特性, 为系统地学习使用 HTML5+CSS3 设计网页奠定基础。

本章的学习目标:

- 了解 Web 网页设计原则
- 了解 HTML5 是如何形成的
- 了解 HTML5 的设计原则
- 了解 HTML5 页面的特征
- 了解网页的基本元素和网页设计的常用技术
- 使用 HTML 创建一些 Web 页面

1.1 Web 网页设计原则

本节介绍开放 Web 标准的重要性、构造良好的语义化标记技术, 以及编写良好的 HTML 是网页设计过程的一部分; 简单、直白的 HTML 结构应当用 CSS 进行样式化等。

1.1.1 Web 标准

成立于 1998 年的 Web Standards Project(Web 标准项目, WSP)一直致力于跨不同浏览器的标准实现和基于标准的 Web 设计方法。其目标是降低 Web 开发的成本与复杂性, 并通过使 Web 内容在不同设备和辅助技术之间更具一致性和兼容性, 提高 Web 页面的可访问性。他们说服浏览器和工具开发商进行改进, 以支持 World Wide Web Consortium(万维网联盟, W3C)推荐的 Web 标准, 如

HTML、CSS 等。由此，Web 标准得到了跨所有主流浏览器的一致实现。

1. 什么是 Web 标准

我们在日常生活中会不经意地用到标准。例如，在买灯泡时，我们知道要买螺旋式或卡口式灯泡，这样才会与家中的灯座装置匹配。标准保证了我们所买的灯泡不会太大，也不会太宽。标准就在我们身边：看看家里的插头、电器的额定功率，以及整个社会都使用的时间、距离、温度度量标准等。

Web 标准出自同样的道理。当浏览器制造商和 Web 开发人员都采用统一的标准时，编写浏览器专用标记的需求就减少了。通过使用结构良好的 HTML 对网页内容进行标记，并使用 CSS 来控制网页的呈现，我们便可设计出能在各种标准兼容浏览器中显示一致的 Web 网站，而不管是什么样的操作系统。重要的是，当同样的标记由基于文本的旧式浏览器或移动设备浏览器呈现时，其内容仍然是可访问的。Web 标准节约了 Web 设计者的时间，让他们坚信自己的杰作是可访问的，而不管用户使用的是哪种平台或哪种浏览器。

2. 使用 Web 标准的好处

使用 Web 标准办法的好处是显而易见的，包括如下方面。

(1) 减少开发时间

只需要建立一个单一的网站，便可以在所有平台、浏览器、设备上运行。如果没有标准，则可能需要为每一种浏览器开发不同的网站。

(2) 创建易于更新与维护的网站

例如，通过 Web 标准和最佳做法，可以更新一个单一的 CSS 文件来改变有若干 HTML 页面的整个网站的样式。在这之前，我们习惯于将样式信息放在 HTML 的内部，这意味着要在每个页面上修改这些信息。这太过烦琐而不便。

(3) 改善搜索引擎排名

HTML 中的内容是基于文本的，因而是搜索引擎可读的。此外，编写良好的副本和恰当使用语义化 HTML(如标题)，可以提高相应关键字的权重，并发送含有 Google Chart 图片的页面。

(4) 改善可访问性

编写良好的 HTML 和 CSS 可以使 Web 站点更易于让不同用户人群访问。例如，残疾人、移动设备用户、使用低带宽连接的人群等。

3. 语义化标记

语义化标记(有时也称为 POSH，代表 Plain Old Semantic HTML)是指要让 HTML 标记侧重于说明内容的含义，例如“这是标题”“这是段落”等。语义化标记不关注内容的表现层含义，如“标题用什么字体、什么颜色、大小如何”等，这种表现层含义属于 CSS 样式要解决的问题。整个 Web 页面的设计过程实际上被分成若干层次，其中一个是标记层，以说明内容的含义，即采用语义化标记来说明内容的含义，这是 HTML 标记的任务。另一个是表现层，用以说明内容的样式或观感，这是 CSS 要解决的问题。

语义化标记是自描述的，用正确的 HTML 元素表示恰当的含义。例如，使用如下标记来表示一个标题：

```
<div id="heading" style="font-size:300%; padding: 10px;">My heading</div>
```

确实，它看起来像一个标题，但它并不是按照标题的含义或目的而起作用的。因此，在搜索引擎优化(标题中的关键字具有更高的权重)、可访问性(屏幕阅读器会使用标题元素作为导航标志)、开发(在使用不当语义元素的情况下，让目标元素带有样式和脚本会需要更多技巧)等方面，都会带来负面影响。

使用适当的元素会好得多，参考如下代码：

```
<h1>My heading</h1>
```

语义化标记也应该尽可能轻量级，也就是要消除所有那些嵌套的<div>标记和其他意大利面条式代码。这可以使文件更小，让编码更加容易。

4. Web 标准层次

五彩缤纷的 Web 页面犹如蛋糕一样，是很多东西的混合体，包括语义化标记、样式信息、脚本/行为等。繁杂的 Web 标准作为整体，是运作整个 Web 世界的一整套合理体系，这也像蛋糕一样，是由很多不同的标准构成的，各种标准都同样重要，不可或缺。其本质是应该将数据结构、样式信息、脚本/行为分离成不同的层次。Web 标准层次包括如下这些。

(1) 数据结构层

语义化 HTML 提供数据结构，是整洁、易于访问的内容集合。HTML5 对此提供很好的支持。应当尽可能使这种数据是可用和可访问的，而无须任何样式方面的脚本强化。

(2) 样式信息层

CSS 提供样式信息，它会利用数据并给出用户所期望的可视化呈现。与 CSS2 相比，CSS3 提供了功能更加强大的工具。

(3) 脚本/行为层

JavaScript(包括在 HTML5 中定义的基本语言和脚本 API)提供脚本/行为层，这为网站增加了可用性和更丰富的功能。

1.1.2 Web 标准与浏览器的兼容性

浏览器的前景正在朝着标准的方向演化和靠拢。Firefox、Safari、Opera、Chrome，当然也包括古老的 IE，都在以不同的速度朝着全面支持 HTML5、CSS3 等标准的方向前进。

但另一个问题是：不仅要支持桌面浏览器。通过移动设备、平板电脑、电视、游戏终端等，在移动过程中访问 Web 的人群快速增长。设备的爆炸式增长，如 Apple 的 iPhone 和 iPad、Google 的 Android 设备、Blackberry、Wii、DS 以及 Philips 和 Sony 的网络电视机等，极大地增加了在移动时、在客厅以及远离桌面的其他地方访问 Web 的人数。

随着访问 Web 内容的设备种类越来越多，精确预测网站在各种用户设备上的外观变得越来越困难。因此，网页的设计不应受像素完美控制的困扰，而是要接纳这种不确定性，设计出可适应不同浏览环境的柔性网站。

1.1.3 可访问性

使用 Web 设计标准会在更广泛的设备上有更好的可访问性，带来更多样的用户等。

设计网页首先要关注不同用户如何使用 Web。有些人使用不同的设备或低速的 Web 连接，有些人只使用键盘，有些人使用屏幕阅读器阅读 Web 页面，有些人听不到音频内容。因此，要熟悉形形色色的 Web 用户。不要只是假设所有人都以相同的模式使用 Web。

HTML5 给 Web 网页的可访问性带来了更广泛的便利。

1.1.4 结构化文档的 Web

可以把 Web 想象成由文档组成的大海，文档之间相互关联，并且与日常生活中遇到的打印文档有着很多相似性。

新闻网站中文章的结构与报纸上文章的结构类似。每一篇文章包含标题、文本段落以及一些图片(有时可能用视频代替图片)。相同点非常明显，而唯一的不同就是：在报纸上，可以在一个版面上放置多篇故事；而在网页上，每篇故事则倾向于独占一个页面。新闻网站还经常使用首页显示新闻头条以及故事梗概。

又比如火车时刻表，火车时刻表的主体是一个表格(table)。从论文资助清单到电视节目表，每一天都会遇到表格化的信息。当这些信息被放到 Web 上时，这些表格就需要被重构。

另一种常见的打印文档是表单(form)。例如，保险公司的一张普通表单包含一些填写区域，用以填写姓名、地址以及保额，同时还有一些复选框，用以指明房子的房间数以及前门锁的种类。Web 上有很多表单，从简单询问寻找事物的搜索框，到在线订购书籍之前需要填写的注册表单。

在上述打印文档与你看到的 Web 页面之间，结构上有着很多相似性。在编写 Web 页面时，HTML 代码会告知 Web 浏览器需要显示的信息结构，即什么文本放在标题、段落或表格中等，从而让浏览器能够恰当地将它们呈现给用户。

1.2 HTML5 概述

2004 年成立的 Web 超文本应用技术工作组(WHATWG)创立了 HTML5 规范，同时开始专门针对 Web 应用开发新的功能。2006 年，W3C 介入 HTML5 的开发，并于 2008 年发布了 HTML5 的工作草案。2009 年，W3C 停止对 XHTML 2.0 的更新。2010 年，HTML5 开始用于解决实际问题。这时各大浏览器厂商开始对旗下产品进行升级以支持 HTML5 的新功能，因此，HTML5 规范得到了持续性的完善。

1.2.1 传统 HTML 与 XHTML

标记语言在我们的日常计算中无处不在。在传统的文字处理文档中，用于描述结构与外观的标记代码常常不是后台编码的。在 Web 文档中，传统 HTML 和 XHTML(XHTML 是使用 HTML 基于 XML 语法规则建立的标记语言)的标记代码是明显可见的。这些非后台的标记语言能够将 Web 页面的结构与外观传递给 Web 浏览器。

从 1991 年底开始推出至今，HTML 和基于 XML 的 XHTML 都发生了许多变化。最早构建 Web 页面的 HTML 版本并没有严格的定义。然而在 1993 年，Internet Engineering Task Force(IETF)开始标准化语言，并且在 1995 年发布了第一个真正意义上的 HTML 标准——HTML 2.0。在以后很长一段

时间里，可能会遇到其他的标记版本，表 1-1 简要介绍了常见的 HTML 和 XHTML 版本。

表 1-1 常见的 HTML 或 XHTML 版本

HTML 或 XHTML 版本	说 明
HTML 2.0	该版本是 Mosaic 等浏览器支持的经典 HTML 版本，支持 HTML 核心元素与功能(例如表格和表单等)，但是不支持较新的高级浏览器功能(例如样式表、脚本、框架等)
HTML 4.0 Transitional	该版本是 W3C 于 1997 年 12 月发布的 HTML 4.0 过渡版，保留了大量 HTML 3.2 中表示外观的元素。该版本提供了向层叠样式表(CSS)，支持与访问多语言环境的元素和特性，以及脚本程序的过渡
HTML 4.0 Strict	该版本为严格的 HTML 4.0 版本，从 HTML 规范中删除了大量表示外观的元素(例如 font)，以支持使用 CSS 的页面格式
HTML 4.0 Frameset	该版本为框架文档提供严格的语法，这在 HTML 以前的版本中是没有实现的
HTML 4.01 Transitional/Strict/Frameset	该版本是对 4.0 标准版的简单升级，仅仅就原始版本中的一些错误进行了纠正
XHTML 1.0 Transitional	该版本是 HTML 规范的 XML 应用的改版。该版本作为过渡版本保留了 HTML 4.0 过渡版的基本外观表示的功能，但是同时向 HTML 应用了严格的 XML 语法规则
XHTML 1.0 Strict	该版本是使用 XML 的 HTML 4.0 严格版本的改版。该版本将所有表示外观的职责都交给 CSS 来完成
XHTML 1.1	该版本是 XHTML 1.0 的改版，其中对标记语言进行了模块化，以方便扩展或缩减。该版本在本书撰写时还没有得到广泛应用，同时并没有对严格的 XHTML 1.0 做大的改变
XHTML 2.0	该版本是对 XHTML 的新的应用，不向后兼容 XHTML 1.0 和传统的 HTML。XHTML 2.0 删除了所有表示外观的标记，并且向语言中引入了不同的新标记和新思想
XHTML Basic 1.0	该版本是在模块化的 XHTML 1.1 基础上设计的，是在功能不太强大的 Web 客户端(例如手机)上使用的版本
XHTML Basic 1.1	该版本是 XHTML 基本规范的改善版本，添加了一些功能，其中一些功能是针对移动设备的界面设计的

1.2.2 HTML5 是如何形成的

HTML5 是 HTML 长期发展过程中的一个里程碑，各种版本的 HTML 以不同规范形成了各种编码风格。尽管它们可能在细节方面有所不同，但它们都有一个共同的基本方面：HTML 是一种标记语言。

1. 超越 HTML 4.0

其实，HTML 4.0 毫无错误之处。它是一个堪称完美的规范，而且其技术与其原先期望要做的工作完全一致：标记出带有相互链接的静态文档。但是 Web 开发人员不会满足于只制作静态文档。他们希望制作具有应用程序行为而不是页面的动态 Web 网站，于是使用了诸如 PHP、JavaScript 以及 Flash 之类的技术。这就形成了要演变的需求。

注意:

Flash 最初变得流行的原因是, 20 世纪 90 年代后期跨浏览器的 Web 标准支持实在太差, 而 Flash 插件提供了一种方式, 使内容能够跨浏览器具有一致的行为。同时, Flash 能够让开发人员在 Web 页面上做一些动画、视频之类的事情。当时, Web 标准还没有支持这些内容的工具。

2. 转向 XHTML 1.0

1998 年, HTML 4.0 规范几近完成, W3C 决定让 Web 转向 XHTML 而不是 HTML, 于是终止了 HTML 4.0 的相关工作(最后一个版本实际上是 HTML 4.01, 它包括一些 bug 修正等), 并全力集中于 XHTML 1.0 规范。

2002 年 8 月, W3C 在考虑 HTML 最佳演化的情况下重磅推出最初的 XHTML。XHTML 1.0 主要是以 XML 词汇重新整理了 HTML 4.0, 使之具备 XML 的那种更严谨的语法规则(例如, 属性值必须有引号、元素需要关闭等)。目标是拥有更好的质量和更有效的标记。

3. XHTML 2.0 的失败

XHTML 的最后一个版本 XHTML 2.0 也是精心编写的规范, 但它完全没有反映出 Web 开发人员在 Web 页面上实际要做的工作, 也未能对 Web 页面上已有的内容做到向后兼容。许多特性的工作方式已有所不同, 例如 XHTML 的 `mimetype(application/xhtml+xml)` 在 IE 中根本不起作用, 而且浏览器中可用的开发人员工具尚未做好使用 XML 的准备。

2004 年, WHATWG(www.whatwg.org) 小组编写了一个更好的标记规范, 使得具有一组能更有效创建新品种 Web 应用程序的特性, 重要的是不会打破向后兼容性。

WHATWG 创建了 Web Application 1.0 规范, 它记录了已有的可交互浏览器行为和特性, 以及用于 Open Web 技术堆栈(如 API 以及新的 DOM 解析规则等)的新特性。经过 W3C 成员之间的多次磋商之后, 2007 年 3 月 7 日, 新的 HTML 工作小组(HTML Working Group, HTML WG)以开放的参与方式重新启动了 HTML 的有关工作。HTML WG 的首要决策之一是, 采纳 Web Application 1.0 规范并称之为 HTML5。

WHATWG 和 W3C 现在以两个不同规范联合开发 HTML5 规范。

- WHATWG HTML 规范是贡献者快速生成和从事创造新特性的地方, 这是在新特性被纳入官方推荐之前的工作方式。注意, 这里去掉了版本号, 表明这是一个活动中的标准, 它会作为单一的整体, 以免除版本化的方式继续进化。
- W3C HTML5 规范是提供最稳定且被广泛认可的特性的地方。该版本描绘了一幅真实画面, 在这里商讨哪些特性是最完整的且最可能在浏览器中得到支持。

4. HTML5 的成功

在 XHTML 2.0 失败之时, HTML5 已经取得了成功, 因为它在开发时考虑到了当前和未来的浏览器开发, 以及过去、现在和将来的 Web 开发任务。

HTML5 是向后兼容的。它包含 HTML 4.0 规范的全部特性, 进行了少量修改和完善。它还包含很多用于建立动态 Web 应用程序以及创建更高质量的标记所需的附加素材, 例如:

- 新的语义化元素, 可以语义化地定义更多标记成分, 而不是使用大量的 class 和 ID。
- 新的元素和 API, 用于为网站添加视频、音频、脚本化图片以及其他内容。