

信息时代数字媒体艺术专业系列教材

网页设计技术

Webpage Design Technology

孙丽娜 编著



北京邮电大学出版社
www.buptpress.com

 信息时代数字媒体艺术专业系列教材

网页设计技术

Webpage Design Technology

孙丽娜 编著



北京邮电大学出版社
[www. buptpress. com](http://www.buptpress.com)

内 容 简 介

本书是数字媒体艺术专业新媒体网络方向的专业教材,内容包括网页前端开发的基础技术:HTML、XHTML、HTML5、CSS2、CSS3、JavaScript 和 jQuery 等。通过细致的规则讲解和案例分析,为网页前端开发的基础技术进行了较为全面的讲解。本书的主要技术相关知识点配有相应的案例讲解,从应用细节进行了说明,使读者能够更好地接受和应用网页前端开发技术,开发设计出优秀的网页作品。

图书在版编目(CIP)数据

网页设计技术 / 孙丽娜编著. --北京:北京邮电大学出版社,2016.8

ISBN 978-7-5635-4813-2

I. ①网… II. ①孙… III. ①网页制作工具 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2016)第 166150 号

书 名:网页设计技术

著作责任者:孙丽娜 编著

责任编辑:满志文

出版发行:北京邮电大学出版社

社 址:北京市海淀区西土城路 10 号(邮编:100876)

发 行 部:电话:010-62282185 传真:010-62283578

E-mail: publish@bupt. edu. cn

经 销:各地新华书店

印 刷:保定市中华美凯印刷有限公司

开 本:787 mm×1 092 mm 1/16

印 张:18.25

字 数:471 千字

版 次:2016 年 8 月第 1 版 2016 年 8 月第 1 次印刷

ISBN 978-7-5635-4813-2

定 价:38.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

编 委 会

名誉顾问：李 杰

主 任：胡 杰

副 主 任：郑志亮 贾立学 陈 薇

编 委：陈超华 马天容 候 明 袁 琳

朱颖博 徐 丹 曾 洁 孙丽娜

丛书总序

数字媒体产业是国家文化创意产业中的重要组成部分，为此，国家十分重视数字媒体教育与专业人才培养。据有关资料统计，截至 2011 年，全国共有 120 余所高校开设了数字媒体艺术专业。数字媒体艺术是一个新专业，它充分体现了 21 世纪数字化生存的细分与融合，体现了艺术与技术的完美结合。如今，美国的动作大片横扫全球，占据了票房的霸主地位，以迪士尼为代表的动画片吸引了数以亿计儿童的眼球；日本、韩国的游戏、动漫产业亦异彩纷呈、蒸蒸日上；处在高速发展中的中国数字媒体产业将上演怎样的精彩呢？

随着移动互联网在全球的蓬勃发展，中国的移动互联网用户数已领先全球，同时国内数字媒体教育正以突飞猛进之势在高速发展。北京邮电大学世纪学院倡导的数字媒体艺术教育依托信息与通信领域，在移动互联网平台上打造数字媒体特色教育，建设与培养从事数字动漫、游戏、影视、网络等数字媒体产品的艺术设计、编创与制作的高级应用型专门人才。

本系列教材编委会依据数字媒体艺术人才培养规律，不断改革创新，精心策划选题，严格筛选课程，准确定位方向。所选编的教材主要涉及动漫、游戏、影视、网络四个领域，重点针对全国各地开设数字媒体艺术专业的本科院校，提供了一套较为完备的、系统的、科学的专业教材。整套教材的主导思路是重视实践案例剖析，强调理论知识积累，教材十分关注数字媒体产业的发展趋势，努力建设特征鲜明的数字媒体艺术教育资源，重视创作理念、艺术技法、科技手段，倡导“理论指导实践、实践反馈于理论”的教学思想。

此次与北京邮电大学出版社合作，正是基于该社鲜明的出版特色，信息通信领域的广泛影响，期望在此基础上全面建设数字媒体艺术的系列教材，为信息产业增添新的特色，为数字媒体教育做出新的贡献。

本套系列丛书主要由北京邮电大学世纪学院数字媒体艺术专业教研团队倾力完成，从教材总体规划、落实选题、整理资料、作者编写、后期修订到编辑出版，凝聚了众多人的心血与热情。作为培养数字媒体艺术人才的一种尝试和探索，难免存在着这样或那样的不足，衷心希望能得到业内各位学者和专家的批评指正。

《信息时代数字媒体艺术专业系列教材》名誉顾问 李杰

前 言

随着信息技术的发展，互联网深入到日常生产生活当中，深刻影响着人们的生活，并引起生产方式的变革。随着“互联网+”时代的来临，网站前端开发的各项技术越来越受到广大 IT 从业人员的重视。网页设计技术是 Web 前端开发的重要内容，Web 前端开发技术包括三个要素：HTML、CSS 和 JavaScript。Web 2.0 时代，Web 标准提出将网页的内容与表现分离；现如今，HTML5 和 CSS3 代表了下一代的 HTML 和 CSS 技术。沿着这样的发展趋势，Web 前端开发的含金量会越来越高，市场对 Web 前端开发人员的需求也应运而生。

本书从 Web 前端开发技术的三个要素着眼，立体地讲解了三个要素的内容，从基础的 HTML、XHTML 和 CSS 2 技术，到应用 JavaScript 和 jQuery 制作网页交互特效，再提高到 HTML5 和 CSS3，通过理论与案例分析相结合的方式，从基础讲解再逐步提高，循序渐进地讲解网页设计技术的相关知识。旨在为有意成为 Web 前端开发设计师的读者提供参考知识。用于教学时，教师应注重 Web 前端开发技术的实践性，有的放矢地增加实践比重，以磨练学生技能的熟练程度。该类知识的学习，仅有理论是不够的，需要应用于实践，才能体现其价值。

网页设计技术的技术性决定了它更加注重应用实践，本书对相应的知识点均提供了案例。然而，书籍容量有限，JavaScript 和 jQuery 的学习还应辅助其他学习资料，进一步系统学习；关于 HTML5 和 CSS3 的新增内容也受篇幅所限未能全面讲解。

本书主要面向数字媒体艺术新媒体网络专业方向的学生，内容易于理解和掌握，也可以为网页设计技术爱好者的学习提供指导和帮助。感谢本书写作过程中给予我帮助的领导、师长和同事，由于个人能力和学时所限，书中不足之处敬请各位读者批评指正。

编 者

目录

CONTENTS

第 1 篇 XHTML+CSS 基础

第 1 章	使用 XHTML 开发网页	3
1.1	HTML 语言/3	
1.2	XHTML 常用元素简介/7	
第 2 章	使用 CSS 定义网页样式	21
2.1	CSS 基本语法/21	
2.2	CSS 盒模型/41	
2.3	CSS 格式化排版/49	
2.4	CSS 布局模型/72	

第 2 篇 JavaScript+jQuery 基础

第 3 章	JavaScript 基础	107
3.1	JavaScript 语法中的基本要求/107	
3.2	JavaScript 核心语法/109	
3.3	JavaScript 对象/115	
3.4	JavaScript 控制动画/120	
第 4 章	jQuery 基础	123
4.1	jQuery 概述/123	
4.2	jQuery 中的事件与动画/127	

第 3 篇 HTML5+CSS3 基础

第 5 章	使用 HTML5 开发网页	135
5.1	什么是 HTML5/135	

- 5.2 HTML5 元素介绍/137
- 5.3 HTML5 多媒体应用/145
- 5.4 HTML5 表单/158
- 5.5 HTML5 Canvas 画布/168

第 6 章 使用 CSS3 定义网页样式

202

- 6.1 功能强大的选择器/202
- 6.2 增强的格式化排版/208
- 6.3 增强的盒模型和盒布局/231
- 6.4 CSS3 动画设计/258
- 6.5 支持多种设备的样式方案/271

参考文献

280

第1篇

XHTML+CSS基础

在Web 1.0时代，表格（table）是网页设计语言中的主流定位方式，进入Web2.0时代，不再采用表格定位技术，而是采用所谓的XHTML+CSS或称DIV+CSS进行定位。XHTML+CSS是网站标准（或称“Web标准”）中常用的术语之一，通常为了说明与HTML网页设计语言中的表格定位方式的区别，因为XHTML网站设计标准中，是采用DIV+CSS的方式实现各种定位。XHTML+CSS网站标准是正在应用的Web标准，目前处于成熟阶段。

第 1 章

使用 XHTML 开发网页

在学习 XHTML 之前，下面先认识一下 HTML 的来龙去脉。

1.1 HTML 语言

1.1.1 什么是 HTML、XML、XHTML

人们习惯用 HTML 页面来称呼所有网页，其中 HTML 表示广义的标识语言，它包括 HTML 和 XHTML。HTML 规范发布的时间点，如图 1-1 所示。

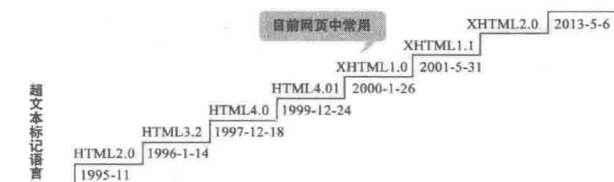


图 1-1 HTML 规范发布时间点

HTML 语言，是用来描述网页的一种语言，指的是超文本标记语言（Hyper Text Markup Language），不是一种编程语言，而是一种标记语言（Markup Language），标记语言是一套标记标签（Markup Tag），使用标记标签来描述网页。

XML 语言，指的是可扩展标记语言（Extensible Markup Language），是适合构建标准网页的语言，也是 W3C（万维网联盟 World Wide Web Consortium 创建于 1994 年，是 Web 技术领域最具权威和影响力的国际中立性技术标准机构。）推荐的最完善的网页结构，但它目前更多的用于 Web 数据的交互，考虑到网页与浏览器的兼容性，W3C 在 XML 基础上发布了一个过渡版本 XHTML。

XHTML 语言，指的是可扩展超文本标记语言（Extensible HyperText Markup Language），与 HTML4.01 几乎是相同的，是更严格、更纯净的 HTML 版本，它兼顾了 XML 用来传输、存储数据的需要和 HTML 用来显示数据的需要，具有如下特点：

- 用户可以扩展元素，从而扩展功能，但在目前版本下，用户只能够使用固定的预定义元素，这些元素基本上与 HTML4 版本元素相同，但删除了描述性元素的使用。
- 能够与 HTML 很好地沟通，可以兼容当前不同的网页浏览器，实现 XHTML 页面的正确浏览。

1.1.2 XHTML 元素、标签和属性

XHTML 网页实际上就是由 XHTML 元素构成的文本文件，任何网页浏览器都可以直

接运行 XHTML 文件。所以, XHTML 元素就是构成 XHTML 文件的基本对象, XHTML 元素可以说是一个统称, XHTML 元素就是通过使用 XHTML 标签进行定义的。

一个 XHTML 元素一般是由一个起始标签和一个结束标签组成(存在单标签), 开始标签使用“<标签名称>”表示, 结束标签使用“</标签名称>”表示(单标签使用“<标签名称 />”表示), 起始标签与结束标签之间包含标签内容。绝大部分的标签都是成对出现的, 如<table></table>、<form></form>。还有少部分单标签, 如
、<hr />等。标签就是用来标记 HTML 元素的。位于起始标签和结束标签之间的文本就是 HTML 元素的内容。

为 XHTML 元素提供各种附加信息的就是 XHTML 属性, 它总是以“属性名=属性值”这种名值对的形式出现, 而且属性总是在 XHTML 元素的开始标签中进行定义。定义属性时, XHTML 建议为属性添加引号, 如果属性本身有双引号, 那建议外面使用双引号, 内部使用单引号。

1.1.3 XHTML 文档结构

为了兼容数以万计的现存网页和不同浏览器, XHTML 页面与 HTML 页面没有太大的区别。但添加了 XML 语言的基本规范和要求。对于初学者, 建议使用 Adobe Dreamweaver 8 或更高版本软件, 在默认情况下可以新建一个标准的 XHTML 网页框架(如代码 1-1 所示)。

代码 1-1 XHTML 网页框架

```
<!-- [XHTML 文档基本框架] -->
<!DOCTYPE html PUBLIC " -//W3C//DTD XHTML1.0 Transitional//EN " http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><!-- 定义 XHTML 文档类型 -->
<html xmlns = "http://www.w3.org/1999/xhtml"><!-- XHTML 文档根元素, 其中
xmlns 属性声明文档命名空间 -->
<head><!-- 头部信息结构元素 -->
<meta http-equiv = "Content-Type" content = "text/html; charset = utf - 8"/>
<!-- 设置文档字符编码 -->
<title>无标题文档</title><!-- 设置文档标题 -->
</head>
<body><!-- 主体内容结构元素 -->
</body>
</html>
```

XHTML 代码不排斥 HTML 规则, 在结构上基本相似, 但如果仔细比较, 会发现 XML 规范的影子, 具体如下:

1. 定义文档类型

在 XHTML 文档第一行新增了<!DOCTYPE>元素, 该元素用来定义文档类型。DOCTYPE 是 document type (文档类型) 的缩写, 它设置 XHTML 文档的版本。使用时应注意:

- 名称和属性必须大写;

• DTD, 表示文档类型定义, 包含了文档的规则, 浏览器通过 DTD 解析页面元素, 并把这些元素所组织的页面显示出来。

没有正确的 DOCTYPE, 页面内的元素和 CSS 不能生效。

XHTML1.0 提供了三种 DTD 文档类型:

(1) 过渡型 (Transitional): 要求非常宽松的 DTD, 它允许继续使用 HTML4 的元素, 但要符合 XHTML 的语法要求, 代码如下:

```
<!DOCTYPE htmlPUBLIC "-//W3C//DTD XHTML1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

(2) 严格型 (Strict): 要求严格的 DTD, 不能使用任何描述性的元素和属性, 代码如下:

```
<!DOCTYPE htmlPUBLIC "-//W3C//DTD XHTML1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

(3) 框架型 (Frameset): 针对框架页面设计的 DTD, 如果页面包含框架, 需用使用这种 DTD, 代码如下:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

对于大多数标准网页设计师来说, 过渡型 DTD (XHTML1.0 Transitional) 是比较理想的选择。因为这种 DTD 允许使用描述性的元素和属性, 也比较容易通过 W3C 的代码校验。

2. 声明命名空间

在 XHTML 文档根元素中必须使用 xmlns 属性声明文档的命名空间。

xmlns 是 XHTML Name Space 的缩写, 中文翻译为命名空间。命名空间是收集元素类型和属性名字的一个详细 DTD, 它允许通过一个 URL 地址指向来识别命名空间。

由于 XML 允许任何人定义自己的标签, 自定义的元素表示什么意思, 是否与他人定义同名标签发生冲突, 这都是命名空间要解决的技术问题。

XHTML 是 HTML 向 XML 过渡的标识语言, 它需要符合 XML 规则, 因此也需要定义命名空间。又因为 XHTML1.0 还不允许用户自定义元素, 因此它的命名空间都相同, 就是 `http://www.w3.org/1999/xhtml`。这就是为什么每个 XHTML 文档的 xmlns 值都相同的缘故。

1.1.4 XHTML 基本语法

XHTML 是根据 XML 语法简化而成, 因此它遵循 XML 文档规范。下面是 XHTML 语言的基本语法:

- XHTML DTD 定义了强制使用的 HTML 元素。所有 XHTML 文档必须进行文件类型声明。在 XHTML 文档中必须存在 html、head、body 元素, 而 title 元素必须位于在 head 元素中。

- 在 XHTML 中, <html> 标签内的 xmlns 属性是必须的。

- 所有的标签必须是闭合的。在 XHTML 中单标签需要用 “/” 闭合, 如
, 成对的标签必须关闭, 如 <p></p>。

- 属性名称必须小写。XHTML 区分大小写, <title> 和 <TITLE> 表示不同的标签。

- 属性值必须加引号。在 HTML 中可以不给属性加引号，但 XHTML 中必须加引号。
- 属性不能简写，必须被赋值，没有值的属性用自身赋值。例如：

错误的写法：

```
<img src>
```

正确的写法：

```
<img src = "src">
```

- 所有标签必须合理嵌套。XHTML 要求严谨的结构，因此嵌套必须按顺序。例如：

错误的写法：

```
<div><p></div></P>
```

正确的写法：

```
<div><p></P></div>
```

- 不要在注释内容中使用“——”。例如：

错误的写法：

```
<!-- 注释 ----- 注释 -->
```

正确的写法：

```
<!-- 注释——注释 -->
```

- 用 id 属性代替 name 属性。

1.1.5 XHTML 元素分类

XHTML 文档是由许多不同的元素组成，而根据这些元素的显示状况，大体可以分为块状元素、内联元素和可变元素三类。下面将详细介绍这三类元素的特点及各种类型中常见的元素。

1. 块状元素 (block element)

顾名思义，块状元素在网页中就是以块的形式显示，所谓块状就是元素显示为矩形区域，CSS3 开始支持定义圆角矩形区域显示，常用的块状元素包括 div、h1~h6、p、table、ul 等。

块状元素一般作为其他元素的容器，它可以容纳内联元素和其他块状元素。在默认情况下，块状元素排斥同行，两个相邻块状元素不会出现并列显示的现象，会独占一行。在默认状态下，块状元素会按顺序自上而下排列，用 CSS 可以改变这种分布形式，而且块状元素都可以定义自己的宽度和高度。

2. 内联元素 (inline element)

inline element 可翻译为内嵌元素、行内元素、直进式元素等。内联元素是基于语义级 (semantic) 的基本元素。任何不是块状元素的可见元素都可以称为内联元素。其表现的特性就是行内布局的形式，也就是说其表现形式是行内逐个显示。

内联元素不会排斥同行其他元素，没有自己的形状，不能定义它的宽、高，它随内容的形状变化而变化，如 span、a、img 等。

块状元素和内联元素是两种基本元素，可以利用 CSS 来改变元素的默认状态。

3. 可变元素

可变元素是根据上下文关系来确定元素是以块状显示还是以内联显示。常见的可变元素

包括：applet（java 小程序）、button（按钮）、map（图像映射）等。

1.2 XHTML 常用元素简介

在了解了 XHTML 的文档结构之后（可参考 1.1.3 小节 XHTML 文档结构），本节介绍 XHTML 文档的常用元素，包括文档头元素和常用 XHTML 标签。

1.2.1 文档头元素

XHTML 文档头部常用元素有文档类型声明<!DOCTYPE>（可参考 1.1.3 小节 XHTML 文档结构）、文档头标签<head>、文档默认地址标签<base>、文档链接<link>、网页标题<title>和元信息标签<meta>。

1. 文档头标签<head>

<head>标签用于定义文档的头部，它是所有头部元素的容器。<head>中的元素可以引用脚本、指示浏览器在哪里找到样式表、提供元信息等等。所有浏览器都支持<head>标签。

文档的头部描述了文档的各种属性和信息，包括文档的标题、在 Web 中的位置以及其他文档的关系等。绝大多数文档头部包含的数据都不会真正作为内容显示给浏览者。

一个简单的 HTML 文档，带有最基本的必须的元素（代码 1-2）：

代码 1-2 简单的 HTML 文档

```
<html>
<head>
<title>文档的标题</title>
</head>
<body>
文档的内容 .....
</body>
</html>
```

2. 文档默认地址标签<base>

<base>标签为页面上的所有链接规定默认地址或默认目标。所有浏览器都支持<base>标签。

在通常情况下，浏览器会从当前文档的 URL 中提取相应的元素来填写相对 URL 中的空白。使用<base>标签可以改变这一点。浏览器随后将不再使用当前文档的 URL，而使用指定的基本 URL 来解析所有的相对 URL。这其中包括<a>、、<link>、<form>标签中的 URL。实例如代码 1-3 所示及表 1-1、表 1-2 所示。

代码 1-3 <base>标签应用示例

```
...
<head>
<base href = "http://www.ccbupt.cn"/>
```

```
<base target = "_blank"/>
</head>
<body>
<img src = "eg_smile.gif"/>
<a href = "http://www.ccbupt.cn">北京邮电大学世纪学院</a>
</body>
...
```

表 1-1 <base> 标签必需的属性

属性	值	描述
href	URL	规定页面中所有相对链接的基准 URL

表 1-2 <base> 标签可选的属性

属性	值	描述
target	_ blank _ parent _ self _ top framename	在何处打开页面中所有的链接： _ blank：在新窗口中打开链接 _ parent：在父窗体中打开链接 _ self：在当前窗体打开链接，此为默认值 _ top：在当前窗体打开链接，并替换当前的整个窗体（框架页） Framename：在对应框架页中打开

3. 文档链接<link>

<link> 标签定义文档与外部资源的关系。<link> 标签最常见的用途是链接样式表。在用于样式表时，<link> 标签得到了几乎所有浏览器的支持。但是几乎没有浏览器支持其他方面的用途。

链接一个外部样式表：

```
<head>
<link rel = "stylesheet" type = " text/css" href = "test. css"/>
</head>
```

4. 网页标题<title>

<title> 元素可定义文档的标题。所有浏览器都支持<title> 标签。

浏览器会以特殊的方式来使用标题，并且通常把它放置在浏览器窗口的标题栏或状态栏上。同样，当把文档加入用户的链接列表或者收藏夹或书签列表时，标题将成为该文档链接的默认名称。

一个简单的 XHTML 文档，带有尽可能少的必需的标签（代码 1-4）：

代码 1-4 <title> 元素应用示例

```
<html>
<head>
<title>XHTML 文档标题</title>
</head>
<body>
```



```
正文内容 .....  
</body>  
</html>
```

5. 元信息标签<meta>

<meta>元素可提供有关页面的元信息（meta-information），比如针对搜索引擎和更新频度的描述和关键词。所有浏览器都支持<meta>标签。

<meta>标签位于文档的头部，不包含任何内容。<meta>标签的属性定义了与文档相关联的名称或值。<meta>标签永远位于 head 元素内部。元数据总是以名称或值的形式被成对传递的，如表 1-3、表 1-4 所示。

表 1-3 <meta>标签必须的属性

属性	值	描述
content	some_text	定义与 http-equiv 或 name 属性相关的元信息

表 1-4 <meta>标签可选的属性

属性	值	描述
http-equiv	content-type expires refresh set-cookie	把 content 属性关联到 HTTP 头部
name	author description keywords generator revised others	把 content 属性关联到一个名称
scheme	some_text	定义用于翻译 content 属性值的格式

常用<meta>标签属性：

字符集声明：每个页面都具有字符集声明，它是非常重要的，它决定了页面文件的编码方式，设定页面使用的字符集，用以说明主页制作所使用的文字已经语言，浏览器会根据此来调用相应的符集显示页面内容。

XHTML 字符集声明：

```
<meta http-equiv = "Content-Type" content = "text/html; charset = utf - 8"/>
```

页面关键词：每个网页应具有描述该网页内容的一组唯一的关键字。

使用人们可能会搜索，并准确描述网页上所提供信息的描述性和代表性关键字及短语。标记内容太短，则搜索引擎可能不会认为这些内容相关。另外标记不应超过 874 个字符。

```
<meta name = "keywords" content = "HTML,XHTML,Javascript"/>
```

页面描述：每个网页都应有一个不超过 150 个字符且能准确反映网页内容的描述标签。

```
<meta name = "description" content = "150 words"/>
```

搜索引擎索引方式，robotterms 是一组使用逗号（,）分割的值，通常有如下几种取值：