

第3章

HTML和XML基础

【本章导读】

在 Web 中,超文本标记语言(HTML)或可扩展标记语言是 W3C 发布的用于进行网页书写的标准和规范。虽然都称为标记语言,但两者的功能完全不同,HTML 规范是一种数据展示技术,它定义了一组固定的标记,来规范网页内容在浏览器中的显示形式。XML 规范则是一种数据表达技术,它的重点是解决数据的逻辑结构和语意表达,以实现文档的结构化和数据处理的自动化。目前,XML 文档数据的显示通常还要转换为 HTML 格式来显示。

本章将从广义标记语言的概念出发,介绍标记语言的概念和功能,对 HTML 规范进行总结,并结合 Web 中一些典型的网页,介绍 HTML 规范中的主要标记的功能及用法。在 XML 相关技术方面,重点讲解 XML 元语言规范、文档类型定义和 XML Schema 技术的思想和出发点,同时讲解 XML 扩展样式语言,以及简要介绍 XML 路径语言 XPath,使大家对这些技术规范的功能及相互关系有一个比较深入的了解,为 Web 应用和开发打下坚实的基础。

【知识要点】

3.1 节: 标记语言的概念、标准通用标记语言(SGML)、超文本标记语言(HTML)、可扩展标记语言(XML)。

3.2 节: 超文本标记语言(HTML)文档结构,标记,标记属性,段落、字体标记,图片标记,超链接标记,表格标记,表单标记,输入域标记,CSS 技术,样式表,CSS 选择符,图层标记,区段标记,帧标记,浮动帧,脚本语言标记。

3.3 节: 可扩展标记语言(XML)、XML 文档结构、文档声明、处理指令、XML 解析器。

3.4 节: 文档类型定义(DTD)、元素、元素声明、基本元素、复杂元素、属性、属性声明、内置属性类型、实体、实体定义、外部 dtd 文件。

3.5 节: XML Schema(模式)、命名空间、简单数据类型、简单数据类型定义、复杂数据类型、复杂数据类型定义、元素声明、属性声明、W3C CSD 内置数据类型、预定义元素、模式文件。

3.6 节: XML 文档对象模型(DOM)、可扩展样式语言(XSL)、模板、XML 路径语言 XPath、路径表达式、路径步、谓词、XPath 轴、XPath 函数。

3.7 节: XML Spy 集成开发环境、dtd 文档的创建、模式文档的创建、XML 实例文档的创建、系统建模、有效性验证。

3.1 标记语言及其发展

广义上的标记语言可以理解为对内容进行描述的规范或标准。例如,在出版印刷行业,编辑人员在进行文档内容编辑时对内容所做的标记,通过这些标记符号来表达对内容的修改信息。在 Web 中,超文本标记语言则通过标记来定义内容在浏览器中的显示样式。随着语义 Web 的发展以及异构环境下的数据交换需求,可扩展标记语言则主要用于对数据结构和数据内容进行标记,成为重要的数据表达、交换和集成标准,对数据的显示则通过其他相关语言来完成,实现了数据内容和显示的分离。

3.1.1 标准通用标记语言

标准通用标记语言(SGML)是一个用来定义在电子表格中如何对文件的结构和内容进行描述的国际标准(ISO-8879)。时间可以追溯到 1969 年,当时美国 IBM 公司的研究人员开始设计一种名为 GML(Generalized Markup Language,通用标记语言)的语言,在印刷、统计等需要大规模数据处理的行业和部门的支持下,这项研究工作持续了十几年,于 1980 年推出了 SGML,并于 1986 年获得国际标准化组织(ISO)的批准。其后,SGML 的发展较为平稳,并不为其领域之外的人们所广泛了解。直至 1991 年,当超文本标记语言(HTML)问世之后,人们才开始认识 SGML。

为了满足各种不同的页面表达需要,SGML 设计得非常复杂,SGML 的正式规范达 500 多页,因此使用起来很不方便,使得它未能得到普及和大规模的应用。虽然 SGML 没有被广泛应用,但是 SGML 定义了标记语言的基本概念,奠定了标记语言发展的技术基础。

现在,在 Web 中普遍应用的 HTML 和 XML 都是在 SGML 的基础上开发成功的,可以说它们都是 SGML 的一个子集。作为互联网信息共享的技术规范,标记语言对互联网的发展起到了巨大的推动作用。

3.1.2 超文本标记语言

超文本标记语言(HTML)起源于标准通用标记语言(SGML),由世界上最大的粒子物理研究实验室欧洲核子研究中心(CERN)于 1991 年首先提出,是推动 Web 迅速发展的原动力,被誉为互联网发展历史上的第一个里程碑。

在互联网发展的早期,为了在各种网络环境之间、不同文件格式之间进行交流,在 SGML 基础上,CERN 提出了超文本标记语言(HTML)的概念。HTML 是一种用来制作超文本文档的简单标记语言,它定义了一组标记符号(tag),对文件的内容进行标注,指出内容的输出格式,如字体大小、颜色、背景颜色、表格形式、各部分之间逻辑上的组织等,从而实现了文件格式的标准化。简单地说,HTML 文件包含了文档数据和显示样式两部分,其中文档数据是显示在 Web 浏览器中的数据内容,显示样式则规定了这些内容在浏览器中以何种格式、样子呈现给用户。通过统一使用支持 html 的浏览软件,用户可以在任意异构的网络环境中,阅读同一个文件,得到相同的显示结果,并可以对文件进行跳跃式阅读,展现了很强的表现力。

HTML 主要版本和发布时间如下。

(1) HTML 2.0, Internet 工程任务组中的 HTML 工作组开发完成了 HTML 2.0, 于 1996 年发布。

(2) HTML 3.2, W3C 于 1997 年 1 月 14 日将其列为推荐版本, 在 HTML 2.0 标准中添加了诸如字体、表格、Java 程序、浮动、上标、下标等特征。

(3) HTML 4.0, W3C 于 1997 年 12 月 18 日将其列为推荐版本, 第二个稍作修正的 HTML 4.0 版本于 1998 年 12 月 24 日发布。HTML 4.0 中最重要的特征是引入了样式表 (CSS) 技术, 使网站样式与内容分离, 使得网站结构更加清晰, 内容更加简洁。

(4) HTML 4.01, W3C 于 1999 年 12 月 24 日将其列为推荐版本, 是 HTML 4.0 的升级版本, 它对原版本做出了部分修正。

(5) HTML 5.0, 作为下一代超文本标记语言的标准, 草案最早于 2004 年提出, 目前仍处于开发阶段, W3C 组织预测, 其完成时间预计要等到 2012 年以后或更长的时间。

在下一代 HTML 标准 HTML 5.0 的发展过程中, W3C 组织希望净化 XHTML 2.0, 回归第一版 HTML 的设计理念。但是, 这样的设计理念遭到了 W3C 之外的一些重要的 HTML 专家, 包括浏览器厂商、Web 开发人员、作者和其他有关人员的质疑, 2004 年, 他们成立了一个独立的工作组, 即 WHATWG (Web 超文本应用程序技术工作组, Web Hypertext Application Technology Working Group), 为新的 HTML 版本提出了新的设计方向。

3.1.3 可扩展 HTML

在 HTML 的发展过程中, 暴露出一些影响其发展的缺陷, 例如: HTML 的标记固定, HTML 只是一种表现技术, 不能表达语义; 不能适应现在越来越多的网络设备和应用的需要, 比如手机、PDA、信息家电都不能直接显示 HTML; 由于 HTML 代码不规范、臃肿, 浏览器需要足够智能和庞大才能够正确显示 HTML; 数据与表现混杂, 页面要改变显示, 就必须重新制作 HTML。为此, W3C 不再继续开发 HTML。

2000 年底, W3C 制定了可扩展 HTML, 即 XHTML, 它是 HTML 向 XML 过渡的一个桥梁。2000 年 1 月 20 日发布 XHTML 1.0, XHTML 1.0 是 HTML 4.01 基于 XML 的形式。2002 年 8 月 5 日, 发布 XHTML 2.0 的第一个工作草案, 其最大的特点就是取消了向后兼容性, 去除了原先版本中的一些标记。例如, 不再支持使用很少的 `` (强调) 和 `` 标记等, 这就使得原先的一些网页在 XHTML 2.0 规范的浏览器中不能正确显示。

需要说明的是, 在 XHTML 的研发过程中, 不确定的东西还很多, 作为 HTML 向 XML 的一种过渡技术, 唯一确定的就是它要更好地表达文档中的语义和结构, 将文档的内容和表现技术能够更好地分离, 更好地实现互联网中的数据交换和展示。

3.1.4 可扩展标记语言

HTML 的不足推动了 XML 的产生和发展, 其核心思想是实现数据和显示的分离。1998 年 2 月 10 日, XML 工作组正式向 W3C 提交了 XML 的最终推荐标准, 这就是 XML

1.0 标准。XML 规范定义了标记语言的主要特征,例如 DTD、XML Schema 等基本要素,这些要素可以很好地用于定义数据,实现异构环境下的数据交换。

对 XML 文档内容的显示、查询及操作则通过其他一系列的规范来实现,这些相关的规范包括可扩展样式语言(XSL)、XML 路径语言(XPath)、XML 查询语言(XQuery)、可扩展连接语言(XLL)以及 XML 文档对象模型(DOM)与简单应用程序接口(SAX)等,通过这些规范来实现对 XML 文档的显示及其他各种操作。

3.2 超文本标记语言

超文本标记语言(HTML)是在 SGML 基础上发展起来的,是互联网中应用最为广泛的标记语言,被称为 World Wide Web 的通用出版语言。在互联网中,绝大多数的网页都是通过 HTML 标记语言书写的,所有的 Web 浏览器都支持 HTML 规范,并能很好地显示 HTML 网页文件。即使是 XML 文档,也通常需要通过 XSLT 转化为 HTML 格式进行显示。

3.2.1 HTML 标记语法和文档结构

HTML 文档是纯文本文件,由“显示内容”和“标记”两部分组成。标记描述显示内容以何种形式在浏览器中显示,也就是说“标记(Tag)”是对内容的标记。标记封装在由小于号(<)和大于号(>)构成的一对尖括号之中,标记一般分首标记和尾标记,它们成对出现。首标记用于开启某种形式的显示,尾标记用于关闭首标记开启的显示功能。例如:<u>欢迎光临</u>,首标记<u>开启下划线功能,尾标记</u>关闭下划线功能。该语句在浏览器中将把文本串“欢迎光临”加上下划线显示。

1. 标记类型

标记分为“单标记”和“双标记”两种类型。“单标记”是指只需单独使用就能完整地表达意思的一类标记,单标记不标记任何内容。这类标记的语法是:

```
<标记>
```

常用的单标记有换行标记
、水平线标记<hr>等。

另一类标记称为“双标记”,由“首标记”和“尾标记”两部分构成,必须成对使用。首标记告诉 Web 浏览器从此处开始执行该标记所表示的功能,而尾标记告诉 Web 浏览器在这里结束该功能。首标记名称前加一个斜杠(/)即成为尾标记。这类标记的语法是:

```
<标记>标记内容</标记>
```

其中“标记内容”部分就是要被这对标记施加作用的部分。例如,如果需要标记一段文本要红色显示,则将文本放在双标记...中即可,即:

```
<font color = "#FF0000">您好</font>
```

标记可以被 Web 浏览器解释,对所标记的内容以特定的样式在浏览器中显示。对于其他的文本阅读器,例如记事本等,则不能解析标记的含义。

2. 标记属性

在 HTML 规范中,标记都设定了默认的显示样式,此外,标记通常还有相应的属性。标记属性分为一般属性和事件属性两种类型,一般属性对应一个相应的属性值,事件属性则对应一段程序代码,当标记上的对应事件发生时,激活相应的事件对应的程序。

设置属性标记的一般形式是:

```
<标记 属性 = "属性值|程序" 属性 = "属性值|程序" ...>
```

各属性之间无先后次序,属性之间用空格分开。属性也可省略(即取默认值),属性值两侧为西文双引号(""),可以使用西文单引号('),或省略不写。

例如:单标记<hr>表示在文档当前位置画一条水平线,默认情况下是从窗口中当前行的最左端一直画到最右端。另外,<hr>标记还有 size、align、width 等属性,其中 size 属性定义线的粗细,默认为 1; width 属性定义线的长度,可取相对值(整个窗口的百分比),也可取绝对值(屏幕像素点的个数),默认值是 100%。align 属性设置对齐方式,可取 left(左对齐,默认值)、center(居中)和 right(右对齐),要画一条 300 像素宽、居中显示的水平线,可写作<hr width="300" align="center">。

再如,在图片标记中可设置事件属性,写为,其中 onclick 就是事件属性,当在图片上单击时,事件被激活,打开一提示窗口。

3. 文档结构

一个 HTML 文档以<html>标记开始,以</html>标记结束,表示这对标记间的内容是 HTML 文档。HTML 文档分成文件头和文件体两个部分,由相应的标记来区分。HTML 文档总体结构如下:

```
<html >
<head >
    头部信息
</head >
<body >
    文档主体
    (语句部分)
</body >
</html >
```

其中,<head>...</head>之间是文件头,由一系列子标记构成,如定义文档标题的<title>标记等,若不需头部信息则可省略此标记。<body>...</body>之间是文件体,表示正文内容的开始,<body>标记一般不能省略。

3.2.2 文件头标记及子标记

在 HTML 文档中,<head>...</head>标记对之间的部分称为文件头。根据 Web 的工作原理,在 Web 服务器和 Web 浏览器的 HTTP 通信中,HTTP 头为浏览器和服务器提供辅助信息,这些辅助信息也可以写在 HTML 文档的头部,为浏览器、搜索引擎等提供信息。例如:设置网页内容字符编码、网页关键字等,这可以使浏览器按照设定的字符编码正

确地显示网页内容,以及让搜索引擎记录网页关键字。

文件头中主要的标记有<title>、<meta>、<script>等,下面分别介绍。

1. <title></title>标记

<title></title>标记出现在<head>...</head>标记对内,用于标识网页主题,其中的内容将在浏览器的标题栏中显示,不出现在页面内。

2. <meta>标记

meta 即“元”的意思,meta data 即元数据,是关于数据的数据。元标记<meta>是最重要的辅助性标记,往往不引起用户的注意,但是它对于网页是否能够被搜索引擎检索、提高网页在搜索列表的排序起着关键的作用,是一个非常有价值的标记。

<meta>标记为单标记,没有尾标记。<meta>标记共有两个属性,分别是 http-equiv 属性和 name 属性,不同的属性又有不同的参数值,这些不同的参数值实现了不同的网页功能。

1) http-equiv 属性

http-equiv 相当于 HTTP 头,向浏览器传回一些有用的信息,以帮助正确显示网页内容,与之对应的属性值为 content,content 中的内容其实就是各个参数的变量值。meat 标记的 http-equiv 属性语法格式是:

```
<meta http-equiv = "参数" content = "参数变量值">
```

其中 http-equiv 属性主要有以下几种参数。

(1) content-type,设定页面文档类型及使用的字符集。

例如:<meta http-equiv="content-type" content="text/html; charset=GB 2312">,该元标记告知浏览器,文档为 HTML 文档,使用的字符集为 GB 2312。如果是 XML 文档,可以设置 content 属性值为 content="text/xml; charset=GB 2312"。

(2) expires,用于设定网页的到期时间。一旦网页过期,必须到服务器上重新下载。

例如:

```
<meta http-equiv = "expires" content = "Thur, 8 May 2008 18:18:18 GMT">
```

(3) pragma,禁止浏览器从本地计算机的缓存中访问页面内容。

例如:

```
<meta http-equiv = "pragma" content = "no-cache">
```

该种设定访问者将无法使用脱机浏览功能。

(4) refresh,自动刷新并指向新页面。

例如:

```
<meta http-equiv = "refresh" content = "60; url = new.htm">
```

则浏览器将在 60 秒后,自动转到 new.htm。利用该功能,可以显示一个封面提示页面,在若干时间后,再自动转移到其他页面。

如果不设置 URL 项,浏览器则刷新本页,这就实现了 Web 聊天室定期刷新的特性。

(5) window-target, 强制页面在当前窗口以独立页面显示。

例如:

```
<meta http-equiv = "window - target" content = "_ top">
```

可以用来防止别人在框架中调用自己的页面。

2) name 属性

name 属性主要用于描述网页, 与之对应的属性值为 content, content 中的内容主要是便于搜索引擎查找信息和分类信息用的。meat 标记的 name 属性语法格式是:

```
<meta name = "参数" content = "具体的参数值">
```

其中 name 属性主要有以下几种参数值。

(1) keywords, 设置网页的关键字, 用来告诉搜索引擎该网页的关键字是什么。

例如:

```
<meta name = "keywords" content = "E-learning, ontology">
```

(2) description, description 用来告诉搜索引擎网站的主要内容。

例如:

```
<meta name = "description" content = "This page is about E-learning etc. ">
```

(3) author, 标注网页的作者。

例如:

```
<meta name = "author" content = "brion@mail. abc. com">
```

(4) robots, 告诉搜索机器人需要索引的页面有哪些。content 的参数有 all、none、index、noindex、follow、nofollow, 默认值为 all。

3. <link> 标记

<link> 标记定义了文档之间的包含。在 HTML 的头部可以包含任意数量的 <link> 标记, <link> 标记带有很多属性, 下面是一些常用的属性。

(1) type, 用于指定被包含的文件类型。例如, text/css 是指包含一个层叠样式表文件。

(2) rel, 定义 HTML 文档和所要包含资源之间的链接关系, 可能的取值很多, 最为常用的取值是 stylesheet, 用于包含一个固定首选样式表单。

(3) href, 指向被包含资源的 URL 地址。

例如, 如果文档包含一个外部的 css 文件, 在文档头部应该定义如下:

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

4. <base> 标记

<base> 标记定义了文档的基础 URL 地址, 在文档中所有的相对地址形式的 URL 都是相对于这里定义的 URL 而言的。一篇文档中的 <base> 标记不能多于一个, 必须放于头部, 并且应该在任何包含 URL 地址的语句之前。

<base>标记包含如下属性。

1) href 属性

href 属性,必选属性,指定了文档的基础 URL 地址。例如,如果希望将文档的基础 URL 定义为 `http://www.abc.com`,则可以使用如下语句:

```
<base href = "http://www.abc.com">
```

当定义了基础 URL 地址之后,文档中所有引用的 URL 地址都从该基础 URL 地址开始,例如,对于上面的语句,如果文档中一个超级链接指向 `gsl/welcome.htm`,则它实际上指向的是如下 URL 地址: `http://www.abc.com/gsl/welcome.htm`。

2) target 属性

target 属性同框架一起使用,它定义了当文档中的链接被单击后,在哪一个框架中打开页面。如果文档中超级链接没有明确指定打开页面的目标框架,则使用这里定义的地址代替。常用的 target 的属性值有:

- (1) `_blank`,表明在新窗口中打开链接指向的页面。
- (2) `_self`,在当前文档的框架中打开页面。
- (3) `_parent`,在当前文档的父窗口中打开页面。
- (4) `_top`,在链接所在的完整窗口中打开页面。

例如: `<base target = "_blank">` 表明页面上所有的链接都在新窗口打开。

5. 背景音乐标记<bgsound>

在文档头内,还可以定义背景音乐,标记为<bgsound>,用以插入背景音乐,可插入的音频文件类型有 Wave 文件(*.wav)、Midi 序列文件(*.midi)、Real Audio 文件(*.ram、*.ra)、AIFF 声音文件(*.aif、*.aifc、*.aiff)、AU 声音文件(*.au、*.snd)。

<bgsound>标记为单标记,一般形式是:

```
<bgsound src = " " autostart = "" loop = "">
```

标记的属性有:

- (1) src 属性,给出音乐文件的 url 值。
- (2) autostart 属性,设置音乐文件播送结束后的处理,如果为 true,则自动播放音乐,为 false 则结束不播放,默认值为 false。
- (3) loop 属性,设置是否自动反复播放,loop=2 表示重复两次,infinite 表示重复多次。

3.2.3 文档体标记及其属性

在<body>...</body>标记对之间的部分称为 html 文档的文档体。文档体中描述的是浏览器窗口中显示的内容。无论网页多复杂,它们都是由文本、图片、超链接等内容构成的,这些页面内容都由相应的 HTML 标记来标记,它们都是<body>标记的子标记。

在讲解具体文档内容标记以前,先介绍文档体标记<body>。<body>标记是一个非常重要的标记,含有大量的属性,许多重要的网页功能都是通过<body>标记的属性实现的。

1. 一般属性

<body>标记的一般属性用于页面的一般性设置,见表 3-1。

表 3-1 <body> 标记一般属性

属 性	用 途	举 例
bgcolor="#rrggbb"	设置页面背景颜色	<body bgcolor="red">, 红色背景
background	设置页面背景图片	background="images/bg1.jpg"
bgproperties	设置成 fixed, 则背景图案不滚动	bgproperties=fixed
topmargin、leftmargin、 bottommargin、rightmargin	设置页面内容的上、下、左、右边距, 像素值	<body leftmargin="0" topmargin="20">
text="#rrggbb"	设置文本颜色	<body text="#ff0000">, 红色文本
link="#rrggbb"	设置未阅读过的超文本链接颜色, 默认值是蓝色	<body link="blue">, 链接为蓝色
vlink="#rrggbb"	设置阅读过的超文本链接颜色, 默认值是紫色	<body vlink="#ff0000">, 红色
alink="#rrggbb"	设置动作中的超文本链接颜色, 默认值是紫色	<body alink="yellow">, 设为黄色

颜色的设置可以通过 HTML 所给定的颜色常量名, 或者 RGB(红、绿、蓝三色的组合) 设置, 例如"#ff0000"表示红色。各个属性可以结合使用, 如: <body bgcolor="red" text="#0000ff">, 设置网页的背景色为红色(red), 文本为蓝色("#0000ff")。

2. 事件属性

当一个 Web 文档被加载显示或者退出(关闭), 当进行移动窗口或改变文档窗口大小等操作时, 会发生相应的事件, 这些事件在<body>标记中通过事件属性来表达。<body>标记常见的事件属性见表 3-2。

表 3-2 <body> 标记中的事件属性

事 件	触 发 条 件	事 件	触 发 条 件
onLoad	页面下载完成时触发	onDbClick	鼠标双击时触发
onUnload	退出页面时触发	onMouseDown	鼠标按键被按下时触发
onFocus	页面窗口获得焦点时触发	onKeyDown	键被按下时触发, 按键的 ASCII 码值保存在 window.event.keyCode 中
onBlur	页面窗口失去焦点时触发	onKeyPress	键被按下然后被释放时触发
onResize	窗口改变大小时触发	onKeyUp	键被释放时触发
onScroll	单击滚动条时触发		
onMouseMove	鼠标移动时触发		

在上述事件中, 有些事件是<body>标记特有的, 有些事件可能存在于多个不同的标记中。

事件属性的值往往是一个 JavaScript 函数, 来完成 Web 编程任务。在 FrontPage、Dreamweaver 等工具软件中, 可以通过行为面板(在 FrontPage 2003 中, 对应“格式”→“行为”菜单命令), 可以显示一个标记支持的行为事件, 并且可自动生成简单的行为 JavaScript 代码, 从而减少用户书写程序的工作量, 具体应用参考后面的章节。

【例 3-1】 HTML 标记的概念及认知。

HTML 标记就是标记内容在 Web 浏览器中以特定的格式显示, 而在非 HTML 应用程序中, 则不能按照标记进行显示。

在 Windows 中,用“记事本”程序输入如图 3-1 所示的 HTML 标记。

将上述内容保存为网页文档 exa3-1. htm,然后双击该文档,文档将在浏览器中打开,显示结果如图 3-2 所示。

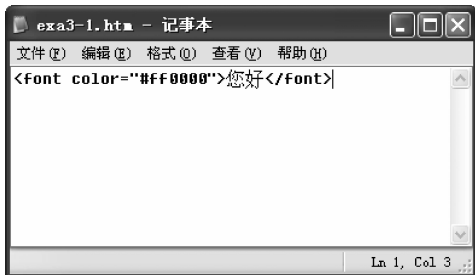


图 3-1 用“记事本”程序创建 HTML 文档



图 3-2 HTML 文档在浏览器中显示界面

可以看到,在浏览器窗口显示红色的“您好”,对比在“记事本”中的显示,可见标记就是告知浏览器,被标记的内容要以什么方式来显示。

3.2.4 文本标记

在一个网页中,文本内容通常是网页的主要内容,在 HTML 规范中,对文本内容的标记较多,它们标记了文本内容在浏览器中的默认显示样式,要修改标记的默认显示,可以通过设置标记属性来完成。HTML 中有关文本的标记及属性见表 3-3。

表 3-3 文本标记及常用属性

分类	标 记	一 般 属 性	事 件 属 性
标题标记	<h1></h1>…<h6></h6>,一级到六级标题标记,对应的字体逐渐渐小	id、style、class、align、title	onclick、ondblclick、onmouseover、onmouseout
文 本 格 式 标记	…,字体标记	id、style、class、face、size、color,设置字体、文字大小和颜色	onclick、ondblclick、onmouseover、onmouseout
	…,粗体标记	id、style、class	同上
	<i>…</i>,斜体标记	同上	同上
	<u>…</u>,下划线标记	同上	同上
回车换行标记	 ,将输出位置转到下一行的开始	id、style、class	无
段落标记	<p>…</p>,段落标记,标记一个段落,输出位置转到下一行开始,并增加一个空行	id、style、class、align、title	onclick、ondblclick、onmouseover、onmouseout
走马灯标记	<marquee>…</marquee>,用于标记一行或多行滚动的文本	id、style、class、height、width,设定滚动字幕的高度和宽度、bgcolor、direction、scrollamount、scrollldlay	同上