



第18章 CSS 3概述

- 18.1 概要介绍
- 18.2 使用CSS 3能做什么



468 ❖ HTML 5与CSS 3权威指南(下册)

从2010年开始,HTML 5与CSS 3就一直是互联网技术中最受关注的两个话题。2010年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Web 1.0的以内容为主的网络,前端主流技术是HTML和CSS;第二阶段是Web 2.0的Ajax应用,热门技术是JavaScript/DOM/异步数据请求;第三阶段是即将迎来的HTML 5+CSS 3的时代,这两者相辅相成,使互联网又进入了一个崭新的时代。

本章将对CSS 3进行一个全面的、概要的介绍,使大家对CSS 3有一个初步的、总体上的认识。

学习内容:

- 掌握CSS 3的基础知识,知道什么是CSS 3,了解CSS 3的发展历史。
- 掌握CSS 3的模块化结构,了解CSS 3中包含了哪些结构。
- 了解CSS 3与CSS 2有什么主要区别,了解CSS 3将对下一代Web平台上的界面设计做出哪些重大贡献。

18.1 概要介绍

18.1.1 CSS 3是什么

首先,我们对CSS 3做一个概要的介绍。什么是CSS 3?CSS 3是CSS技术的一个升级版本,是由Adobe Systems、Apple、Google、HP、IBM、Microsoft、Mozilla、Opera、Sun Microsystems等许多Web界的巨头联合组成的一个名为“CSS Working Group”的组织共同协商策划的。虽然目前很多细节还在讨论之中,但它还是不断地朝前发展着。2010年在HTML 5成为IT界人士关注的焦点的同时,它也开始慢慢地普及开来。

18.1.2 CSS 3的历史

接下来,我们从总体上看一下CSS的发展历史。

□ CSS 1。

1996年12月,CSS 1(Cascading Style Sheets,level 1)正式推出。在这个版本中,已经包含了font的相关属性、颜色与背景的相关属性、文字的相关属性、box的相关属性等。

□ CSS 2。

1998年5月,CSS 2(Cascading Style Sheets,level 2)正式推出。在这个版本中开始使用样式表结构。

□ CSS 2.1。

2004年2月,CSS 2.1(Cascading Style Sheets,level 2 revision 1)正式推出。它在CSS 2的基础上略微做了改动,删除了许多诸如text-shadow等不被浏览器所支持的属性。

现在所使用的CSS基本上是在1998年推出的CSS 2的基础上发展而来的。10年前在Internet刚开始普及的时候,就能够使用样式表来对网页进行视觉效果的统一编辑,确实是一件可喜的事情。但是在这10年间CSS可以说是基本上没有什么很大的变化,一直到2010年终于推出了一个全新的版本——CSS 3。

18.2 使用CSS 3能做什么

18.2.1 模块与模块化结构

在CSS 3中，并没有采用总体结构，而是采用了分工协作的模块化结构，这些模块如表18-1所示。

表18-1 CSS 3中的模块

模块名称	功能描述
basic box model	定义各种与盒相关的样式
Line	定义各种与直线相关的样式
Lists	定义各种与列表相关的样式
Hyperlink Presentation	定义各种与超链接相关的样式。譬如锚的显示方式、激活时的视觉效果等
Presentation Levels	定义页面中元素的不同样式级别
Speech	定义各种与语音相关的样式。譬如音量、音速、说话间歇时间等属性
Background and border	定义各种与背景和边框相关的样式
Text	定义各种与文字相关的样式
Color	定义各种与颜色相关的样式
Font	定义各种与字体相关的样式
Paged Media	定义各种页眉、页脚、页数等页面元数据的样式
Cascading and inheritance	定义怎样对属性进行赋值
Value and Units	将页面上各种各样的值与单位进行统一定义，以供其他模块使用
Image Values	定义对image元素的赋值方式
2D Transforms	在页面中实现2维空间上的变形效果
3D Transforms	在页面中实现3维空间上的变形效果
Transitions	在页面中实现平滑过渡的视觉效果
Animations	在页面中实现动画
CSSOM View	查看管理页面或页面的视觉效果，处理元素的位置信息
Syntax	定义CSS样式表的基本结构、样式表中的一些语法细节、浏览器对于样式表的分析规则
Generated and Replaced Content	定义怎样在元素中插入内容
Marquee	定义当一些元素的内容太大，超出了指定的元素尺寸时，是否以及如何显示溢出部分
Ruby	定义页面中ruby元素（用于显示拼音文字）的样式
Writing Modes	定义页面中文本数据的布局方式
Basic User Interface	定义在屏幕、纸张上进行输出时页面的渲染方式
Namespaces	定义使用命名空间时的语法
Media Queries	根据媒体类型来实现不同的样式
' Reader ' Media Type	定义用于屏幕阅读器之类的阅读程序时的样式
Multi-column Layout	在页面中使用多栏布局方式
Template Layout	在页面中使用特殊布局方式
Flexible Box Layout	创建自适应浏览器窗口的流动布局或自适应字体大小的弹性布局
Grid Position	在页面中使用网格布局方式
Generated Content for Paged Media	在页面中使用印刷时使用的布局方式

那么,为什么需要分成这么多模块来进行管理呢?

这是为了避免产生浏览器对于某个模块支持不完全的情况。如果只有一个总体结构,这个总体结构会过于庞大,在对其支持的时候很容易造成支持不完全的情况。如果把总体结构分成几个模块,各浏览器可以选择对于哪个模块进行支持、对哪个模块不进行支持,支持的时候也可以集中把某一个模块全部支持完了再支持另一个模块,以减少支持不完全的可能性。

例如,台式计算机、笔记本和手机上用的浏览器应该针对不同的模块进行支持。如果采用模块分工协作的话,不仅是台式计算机,各种设备上所用的浏览器都可以选用不同模块进行支持。

18.2.2 一个简单的CSS 3示例

现在,我们已经对CSS 3的模块和模块化结构有了一个初步的认识,那么,究竟我们能够用CSS 3来做些什么呢?

这里,我们通过一个示例,来将CSS 2与CSS 3做一个对比,借此对CSS 3有一个初步的印象。

在这个示例中,我们给页面上的某个div区域添加一个彩色图像边框,这样可以使这个区域看上去漂亮很多,生动很多。

在CSS 2中,当然可以实现这个效果,如代码清单18-1所示。

代码清单18-1 使用CSS 2给div区域添加图像边框

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<style type="text/css">
#image-boarder{
margin:3px;
width:450px;
height:104px;
padding-left:14px;
padding-top:20px;
background:url(test.png);
background-repeat:no-repeat;
}
</style>
</head>
<body>
<div id="image-boarder">
• 示例文字1<br/>
• 示例文字2<br/>
• 示例文字3<br/>
• 示例文字4<br/>
</div>
</body>
</html>
```

这段代码在Firefox浏览器中的运行结果如图18-1所示。

接下来，我们看一下在CSS 3中如何实现这个功能。

在CSS 3中，添加了很多新的样式，譬如可以创建圆角边框，可以在边框中使用图像，可以修改背景图像的大小，可以对背景指定多个图像文件，可以修改颜色的透明度，可以给文字添加阴影，可以在CSS中重新指定表单的尺寸等。

在代码清单18-2中，我们使用CSS 3来实现与代码清单18-1相同的功能。具体操作的时候，只要给页面中的div元素增加一个border-image属性，然后在该属性中指定图像文件、边框宽度与图像拉伸方式就可以了。



图18-1 使用CSS 2样式添加图像边框

代码清单18-2 使用CSS 3给div区域添加图像边框

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<style type="text/css">
#image-boarder{
width:450px;
padding-top:20px;
padding-left:14px;
-moz-border-image:url(test.png) 3 stretch stretch; //指定边框图像
}
</style>
</head>
<body>
<div id="image-boarder">
• 示例文字1<br/>
• 示例文字2<br/>
• 示例文字3<br/>
• 示例文字4<br/>
</div>
</body>
</html>
```

这段代码的运行后结果与图18-1所示结果相同。

虽然目前看来两种方法都达到了同样的效果，只是实现方法不同而已。但是如果再在div中增加一行文字，我们看一下使用CSS 2中的样式表后会是什么情况，如图18-2所示。

同样的，来看一下使用CSS 3中的样式表后会是什么情况，如图18-3所示。

为什么在CSS 3中文字没有超出边框图像之外？这是因为在CSS 3样式表中，在指定边框图像的同时，也指定了图像允许拉伸来自适应div区域的高度，而不是采取CSS 2中将div区域高度设为边框图像高度的方式。那么，也许有人会问，如果在CSS 2的div元素的样式代码中不指定div区域的高度是否可以呢？这样的话就会出现如图18-4所示的情况。

472 ❖ HTML 5与CSS 3权威指南(下册)



图18-2 使用CSS 2样式表,当文字超过图像高度时的页面外观



图18-3 使用CSS 3样式表,当文字超过图像高度时的页面外观



图18-4 在CSS 2的样式代码中不指定div区域高度的效果

从图中可以看出,当只有一行文字的时候,该边框图像又不能完全显示了。因此,当div区域中的文字高度处于不断变化的状态时,使用CSS 2样式表添加边框图像的操作相对来说就比较麻烦。在CSS 3中考虑到了这种情况,添加了允许边框图像自动拉伸的属性,从而解决了这个问题。

关于如何使用border-image这个属性,我们将在后面进行详细介绍。在这里,我们通过这个示例,向大家表明了目前在CSS 2中一些比较难以处理的情况,在CSS 3中通过使用新增属性很容易就能够解决。

这对界面设计来说,无疑是一件非常可喜的事情。在界面设计中,最重要的就是创造性,如果能够使用CSS 3中新增的各种各样的属性,就能够在页面中增加许多CSS 2中没有办法解决的样式,摆脱现在界面设计中存在的许多束缚,从而使整个网站或Web应用程序的界面设计进入一个新的台阶。