# 第三章 CSS3边框

提到边框，大家首先想到的是CSS的border属性。不错，border属性是CSS中盒模型的基础属性之一。在CSS3中，关于边框的运用会有什么样的不同之处呢？又将如何使用呢？本章我们带着这些问题开始我们的CSS3边框之旅。

## 3.1 CSS3边框简介

border属性在CSS1中就已经定义了，使用它可以设置元素的边框风格，例如设置不同的边框颜色以及粗细。在详细介绍CSS3边框的运用之前，先简单回顾边框属性。

##### 3.1.1 边框的基本属性

CSS1和CSS2中的边框属性其实很简单，其主要包括三个类型值。

border-width：设置元素边框的粗细。

border-color：设置元素边框的颜色。

border-style：设置元素边框的类型。

在实际中可以将上面三个属性合并在一起，其缩写的语法：

border:border-width border-style border-color;

缩写后的每个属性之间使用空格隔开，而且他们之间没有先后顺序，可这里三个值中唯一需要的值是“border-style”，因此，这样写边框样式将没有任何效果。

.elm{border:3px red}

此时浏览器将“border-style”解析为“none”。要是这样设置，这时候元素的边框是实线，粗线将是其默认值。

.elm{border:solid}

边框border-width的默认值是“medium”（大约等于3~4px）；border-color的默认颜色就是字体的颜色。

在Web实际制作之中，时常只是为了方便使用，CSS中的border可以给不同的边设置不同的风格，其也遵守“TRBL”原则（Top/Right/Bottom/Left），例如单独写边框类型。

border-top-style:/\*设置元素顶部边框类型\*/

border-right-style:/\*设置元素右边边框类型\*/

border-bottom-style:/\*设置元素底部边框类型\*/

border-left-style:/\*设置元素左边边框类型\*/

上面是边框类型的扩展写法，同样的道理，border-color和border-width也可以像上面一样使用。除了上面的写法之外，还有一种简写形式。

border-style:solid;

/\*一个值时，表示四条边都是solid类型\*/

border-style:solid dotted;

/\*两个值时，第一个值表示元素的上下边框类型，第二个值表示左右边框类型\*/

border-style:solid dotted dashed;

/\*三个值时，第一个值表示元素顶边的类型，第二个值表示左右边框类型，第三个值表示底部边框类型\*/

border-style:solid dotted dashed inset;

/\*四个值时，第一个值表示元素顶边的类型，第二值表示右边框类型，第三个值底部边框类型，第四个值表示左边框类型\*/

同样的原理，border-color和border-width具有一样的使用方法。如果只需要设置元素某部分具有表框效果，我们可以合成起来。

li{

border:1px solid red;

border-width:1px 0;

}

仅两行代码就表达出元素li顶部和底部都有一条1px的红色实线。这样方便维护代码，并提升CSS性能。

##### 3.1.2 边框的类型

CSS中使用border-style为元素border定义边框类型，常见的有实线“solid”、虚线“dashed”、点状线“dotted”等。下面一起看看CSS中的border-style的几种类型效果。

border-style值列表

|  |  |
| --- | --- |
| 属性值 | 功能描述 |
| None | 定义无边框 |
| Hidden | 与“none”相同。不过应用于表时除外，对于表，hidden用于解决边框冲突 |
| Dotted | 定义点状表框 |
| Dashed | 定义虚线边框 |
| Solid | 定义实线边框 |
| Inherit | 规定应该从父元素继承边框样式。部分浏览器不支持这个属性。 |

### 3.1.3 谁在使用CSS3边框

CSS3增强的边框样式具有强大的生命力，灵活使用这些属性可以设计很多优美精巧的UI界面效果。这些属性谁在使用呢？

border-color受制与浏览器兼容性，至今在项目中使用该属性的项目几乎不存在。

border-image浏览器的支持度强一些，但运用在项目中仅存在一些前端爱好者的blog中。

Border-radius得到浏览器的强大支持，在互联网上随处可见。

Box-shadow 目前在web页面上CSS3的盒子阴影特性应用非常广泛。

### 3.2 CSS3边框颜色属性

border-color属性早在CSS1中就已经定义了。不过，CSS3增强了这个属性的功能，使用它可以为元素边框设置更多的颜色，从而方便web设计人员设计出更为绚丽的边框效果，如渐变的边框效果、多颜色的边框效果等。

##### 3.2.1 border-color属性的语法及参数

Border-color的语法看上去和CSS1中完全相同，但是为了避免与border-color属性的原功能发生冲突，CSS3在这个属性上作出了一定的修改。语法如下。

border-color: [<color> | transparent ]{1,4} | inherit

换句话说，如果使用border-color这种缩写语法，将不会有任何效果，必须将这个border-color标准写法拆成四个边框，使用多个颜色才有效果。

Border-top-colors:[<color>|transparent]{1,4} | inherit;

Border-right-colors:[<color>|transparent]{1,4} | inherit;

Border-bottom-colors:[<color>|transparent]{1,4} | inherit;

Border-left-colors:[<color>|transparent]{1,4} | inherit;

由于CSS3的border-color属性还没成为标准规范，为了让不同浏览器能渲染正常，有必要加上前缀

有些情况中，为了使用CSS3属性，可能必须添加4行代码或者更过行。例如，为了让border-color 在 Filefox浏览器下正常，需要加上前缀“-moz-”。

-moz-border-top-colors:#111 #222 #333 #444 #555;

-moz-border-right-colors:#111 #222 #333 #444 #555;

-moz-border-bottom-colors:#111 #222 #333 #444 #555;

-moz-border-left-colors:#111 #222 #333 #444 #555;

现在的规范还不是最终版本，在执行中还会有一些漏洞。因此，在执行这些功能时，使用供应商前缀来提供数值，并且使用无前缀声明来提供每个属性的永久版本。当规范成为最终版本并且经过完善后，浏览器前缀将被取消。例如：

-moz-border-top-colors:#111 #222 #333 #444 #555;

-moz-border-right-colors:#111 #222 #333 #444 #555;

-moz-border-bottom-colors:#111 #222 #333 #444 #555;

-moz-border-left-colors:#111 #222 #333 #444 #555;

-webkit-border-top-colors:#111 #222 #333 #444 #555;

- webkit -border-right-colors:#111 #222 #333 #444 #555;

- webkit -border-bottom-colors:#111 #222 #333 #444 #555;

- webkit -border-left-colors:#111 #222 #333 #444 #555;

-ms-border-top-colors:#111 #222 #333 #444 #555;

- ms-border-right-colors:#111 #222 #333 #444 #555;

- ms-border-bottom-colors:#111 #222 #333 #444 #555;

- ms-border-left-colors:#111 #222 #333 #444 #555;

-o-border-top-colors:#111 #222 #333 #444 #555;

- o-border-right-colors:#111 #222 #333 #444 #555;

- o-border-bottom-colors:#111 #222 #333 #444 #555;

- o-border-left-colors:#111 #222 #333 #444 #555;

border-top-colors:#111 #222 #333 #444 #555;

border-right-colors:#111 #222 #333 #444 #555;

border-bottom-colors:#111 #222 #333 #444 #555;

border-left-colors:#111 #222 #333 #444 #555;

即使用这些前缀来维护代码似乎需要很多工作，现在使用CSS3，仍然是利大于弊。虽然需要改变一些前缀属性来修改设计元素，相对于通过图形软件更改背景图像或处理那些其他标记和hack脚本，维护基于CSS3的设计要容易一些。

Border-color属性的参数其实很简单，就是颜色值<color>,可以为任意合法的颜色值或者颜色值列表。当border-color只设置一个颜色值时，效果和CSS1中的border-color效果一样。只有设置了n个颜色，并且边框也为n像素，就可以使用CSS3的border-color属性设置n个颜色，每种颜色显示1像素的宽度，如果宽度值大于颜色数量的值，最后一种颜色用于显示剩下的宽度。例如，元素的边框设置为20px，而元素的边框颜色只设置了10个，剩下的10px宽度都将显示最后一种颜色。

### 3.3 CSS3图片边框属性

border-image效果在CSS2中，只有使用背景图片来制作，而且制作过程非常复杂，做完后也很难维护。如今CSS3中增添了一个图片边框的属性，能模拟出background-image属性的功能，功能比background-image强大，我们可以使用border-image属性给任何元素设置图片效果边框，还可以使用这个制作圆角按钮效果、渐变的Tabs效果等。

##### 3.3.1 border-image属性的语法及参数

为了能更好地学习和理解border-image这个属性，还是从最基本的语法入手。

Border-image:none | <image> [<number>] | <percentage> {1,4} [/ <border-width> {1,4}] ? [stretch | repeat | round] {0,2}

接下来就给大家说说这些参数的含义与使用方法。

None：默认值，表示边框无背景图片。

<image>：设置背景图片，这跟background-image一样，可以使用绝对或相对的URL地址，来指定边框的背景图片。

<number>：number是一个数值，用来设置边框或者边框背景图片的大小。其单位是像素（px），可以使用1~4个值表示4个方位的值，大家可以参考border-width设置方式。

<percentage>：percenttage也是用来设置边框或者边框背景图片的大小，跟number不同的是，percentage使用的是百分比。

Stretch 、repeat、round ：这三个属性参数是用来设置边框背景图片的铺放方式，类似于background-position，其中stretch会拉伸边框背景图片、repeat是会重复边框背景图片、round是平铺边框背景图片，其中stretch为默认值。

##### 3.3.2 border-image属性使用方法

为了更好地理解，暂时把border-image在语法的表达形式进行属性的分解阐述。

1. 引入背景图片：border-image-source。
2. 切割引入背景图片：border-image-slice。
3. 边框图片的宽度：border-image-width。
4. 边框背景图片的排列方式：border-image-repeat。

接下来重点学习border-image拆分出来的四个属性。

1. **border-image-soure**

**语法：**

**border-image-source:url(image url);**

**/\*image url 可以是以边框图片的相对地址，也可以是绝对地址\*/**

**border-image-source跟CSS2中的background-image属性相似，也是通过url()来调用背景图片，图片的路径可以是相对地址，也可以是绝对地址，当然不想使用背景图片也可以设置为“border-image:none”；其默认值就是none。**

1. **border-image-slice**

**语法：**

**Border-image-slice:[<number> | <percentage> ] {1,4} && fill ?**

**Border-image-slice 是用来分解引入进来的背景图片，这个参数相对来说比较复杂和特别，主要表现在以下几点。**

1. **取值支持<number>|<percentage>。其中number是没有单位的，因为其默认的单位就是像素。除了直接用number来设置外，还可以使用百分比来表示，即相对于边框背景图片而言的，例如边框图片的大小是300px \* 240px ，取百分比为25% ，30%，15% ，20% ，它们实际对应的效果就是裁剪了图片的60px，90px，36px，60px的四边大小。**

Border-image-slice中的number或者percentage都可以取1~4个值，这个类似于CSS2中的border-width的取值方式，也遵从top、right、bottom、left的规则。

Fill从字面上说就是填充的意思，如果使用这个关键字时，图片边界的中间部分将保留下来。

1. 剪切的特性。在border-image中slice是一个关键部分，也是让人难以理解的部分。如果理解CSS3的clip属性，再理解border-image-slice相对会轻松一些。Border-image-slice虽然表面上说不是裁剪，它把通过border-image-source取到的边框背景图片切成九份，再像background-image一样重新布置。

**3. border-image-width**

语法：

Border-image-width:[<length>|<percentage>|<number>|auto]{1,4}

用来设置边框背景图片的显示大小，其实也可以理解为border-width。虽然W3C定

义了border-image-width属性，但各浏览器还是将其视为border-width来用，也就是说它和border的使用方法是一样的。

**4.border-image-repeat**

**语法：**

**border-image-repeat:[stretch | repeat | round]{1,2}**

**用来指定边框背景图片的排列方式，其默认值为stretch。这个属性设置参数和其他的不一样，border-image-repeat不遵循top、right、bottom、left的方位原则，它只接受两个（或一个）参数值，第一个值表示水平方向的排列，第二个值表示垂直方向的排列方式。当只取一个值时，表示水平和垂直方向的排列方向相同。如果你不显示设置任何值时，水平和垂直都会默认以其默认值stretch方式来排列。**

### **3.4 CSS3圆角边框属性**

CSS3中专门针对元素的圆角效果增加了一个圆角属性border-radius。

语法：

Border-radius：none|<length>{1,4}[/<length>{1,4}]?

Border-radius是一种缩写方法。如果反斜杠符号“/”前面的值是设置元素圆角的水平方向半径，“/”后面的值是设置元素圆角的垂直方向半径；如果没有“/”，则元素圆角的水平和垂直方向的半径值相等。

### 3.5 CSS3盒子阴影属性

Box-shadow也是CSS3新增的一个重要属性，用来定义元素的盒子阴影。本节主要介绍CSS3的box-shadow的属性以及如何使用。

3.5.1 box-shadow属性的语法及参数

在具体学习box-shadow使用方法之前，我们必须先知道box-shadow使用的语法规则。

box-shadow:none|[<length> <length> <length> ?<length>? || <color>][,<length> <length> <length> ?<length>? || <color>]+

上面的语法规则可以简写如下：

Box-shadow:none|[inset x-offset y-offset blur-radius spread-radius color], [inset x-offset y-offset blur-radius spread-radius color]

Box-shadow属性可以使用一个或多个投影，如果使用多个投影时必须使用逗号“,”隔开。

其实box-shadow属性很简单，可以为其设置以下参数。

None：默认值，元素没有任何阴影效果。

Inset：阴影类型，可选值。如果不设置，其默认的投影方式是外阴影，如果取其唯一值“inset”，就是给元素设置内阴影。

X-offset：阴影水平偏移量，其值可以是正负值。如果取正值，则阴影在元素的右边，反之取负值，阴影在元素的左边。

Y-offset：阴影垂直偏移量，其值可以是正负值。如果取正直，则阴影在元素的底部，反之取负值，阴影在元素的顶部。

Blur-radius：阴影模糊半径，可选参数。其值只能是正值，如果取值为0时，表示阴影不具有模糊效果，如果取值越大，阴影的边缘就越模糊。

Spread-radius：阴影扩展半径，可选参数。其值可以是正负值，如果取值为正值，则整个阴影都延展扩大，反之取值为负值，则整个阴影都缩小。

Color：阴影颜色，可选参数，如果不设定任何颜色时，浏览器会取默认色，但各浏览器默认色不一样，特别是在Webkit内核下的浏览器将无色，也就是透明，建议不要省略这个参数。

3.5.2 box-shadow属性使用方法

和PSD软件制作图片相比，box-shadow修改元素的阴影效果要方便的多，因为box-shadow可以修改六个参数，得到不同的效果，下面结合一些简单的案例来对box-shadow属性进行演示说明。

1. 单边阴影效果

定义元素的单边阴影效果和协调border的单边边框颜色是相似的，例如：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>box-shadow设置单边阴影效果</title>

<style type="text/css">

.box-shadow{

width:200px;

height:100px;

border-radius: 5px;

border: 1px solid #CCC;

margin: 20px;

}

.top{

box-shadow: 0 -2px 0 red;

}

.right{

box-shadow: 2px 0 0 green;

}

.bottom{

box-shadow: 0 2px 0 blue;

}

.left{

box-shadow: -2px 0 0 orange;

}

</style>

</head>

<body>

<div class="box-shadow top"></div>

<div class="box-shadow right"></div>

<div class="box-shadow bottom"></div>

<div class="box-shadow left"></div>

</body>

</html>

这个案例中，使用box-shadow给元素设置了顶边、右边、底边和左边的单边阴影效果。主要通过box-shadow的水平和垂直阴影的偏移量来实现，其中x-offset为正值时，生成右边阴影，反之为负值时，生成左边阴影；y-offset为正值时生成底部阴影，反之为负值时，生成底部阴影。此例中是一个单边实影投影效果（阴影模糊半径为0），但是如果阴影的模糊半径不是0，上面的方法还能不能实现单边阴影效果呢？不着急来回答，在上面的实例中添加一个模糊半径，例如：

.top{

box-shadow: 0 -2px 5px red;

}

.right{

box-shadow: 2px 0 5px green;

}

.bottom{

box-shadow: 0 2px 5px blue;

}

.left{

box-shadow: -2px 0 5px; orange;

}

这个效果并不是理想的单边阴影效果，当box-shadow添加了5px阴影模糊半径后阴影不在是实体投影，阴影清晰度向外扩散，更具阴影的效果。但造成了另外一个问题，给元素其他三个边加上淡淡的阴影效果，可这并不是设计需要的效果。

那究竟要怎么做呢？此时，box-shadow属性中的阴影扩展半径（spread-radius）会是一个很关键的属性，要实现阴影效果，必须配上这个属性（除单边实影之外）。

.top{

box-shadow: 0 -4px 5px -3px red;

}

.right{

box-shadow: 4px 0 5px -3px green;

}

.bottom{

box-shadow: 0 4px 5px -3px blue;

}

.left{

box-shadow: -4px 0 5px; -3px orange;

}

上面的代码调整了阴影的位移量，新增了box-shadow的扩展半径。

2. 四边相同阴影效果

Box-shadow给元素设置相同的四边阴影效果，其实分为两种，这里先看第一种。

（1）只设置阴影模糊半径和阴影颜色

.box-shadow{

width:200px;

height:100px;

border-radius: 5px;

border: 1px solid #CCC;

margin: 20px;

box-shadow: 0 0 10px #06c;

}

在这个示例基础上，添加box-shadow扩展半径还可以控制阴影深度，如果取正值将加深阴影的深度，如果取负值可以向内压缩阴影，直到扩展半径等于模糊半径时，阴影会完全消失。例如：

.box-shadow{

width:200px;

height:100px;

border-radius: 5px;

border: 1px solid #CCC;

margin: 20px;

box-shadow: 0 0 10px 10px #06c;

}

接下来，将扩展半径改成“-10px”，此时将看不到任何阴影效果。

.box-shadow{

width:200px;

height:100px;

border-radius: 5px;

border: 1px solid #CCC;

margin: 20px;

box-shadow: 0 0 10px -10px #06c;

}

（2）只设置扩展半径和阴影颜色

另一种设置四边相同阴影效果，是设置扩展半径和阴影颜色，先看一下简单的示例。

.box-shadow{

width:200px;

height:100px;

border-radius: 5px;

border: 1px solid #CCC;

margin: 20px;

box-shadow: 0 0 0 10px #06c;

}

上例制作的阴影效果和元素设置“10px”实线边框一样。

Border:10px solid #06c;

如此一来，可以利用box-shadow扩展半径制作类似于边框的效果，但实质上并非边框，因为box-shadow并不是盒模型中的元素，不会计算到内容宽度。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>border与box-shadow</title>

<style type="text/css">

.box{

width:200px;

height:100px;

text-align: center;

line-height: 100px;

float: left;

margin: 30px;

}

.border{

border: 10px solid red;

}

.box-shadow{

box-shadow: 0 0 0 10px red;

}

</style>

</head>

<body>

<div class="box border">Border</div>

<div class="box box-shadow">Box-shadow</div>

</body>

</html>

Box-shadow不会影响页面的任何布局。div.border的边框被计算了宽度，但div.box-shadow的阴影被浏览器忽略不计，所以借助box-shadow属性的这个特性，border-shadow用来模拟元素的边框效果可以自由的使用但必须注意其层级关系

W3C标准规范中描述了box-shadow的工作放式，直观告诉我们box-shadow在元素盒模型中的层次关系，比如说border-radius圆角、阴影扩展、阴影模糊以及padding是如何影响对象阴影的。非零值的border-radius会以相同的作用影响阴影的外形，但border-image不会影响对象阴影的任何外形；对象阴影同盒模型的层次一样，外阴影会在对象背景之下，

内阴影会在边框之下，背景之上。所以整个层级就是：边框在内阴影之上，内阴影在背景图片之上，背景图片在背景色之上，背景色在外阴影之上。

3.内阴影

前几种都是外阴影的使用方法，其实使用inset属性值可以改变元素的阴影类型，将元素的默认外阴影类型重置为内阴影类型。例如：

.box-shadow{

width:200px;

height:100px;

border-radius: 5px;

margin: 30px;

border: solid 3px #CCC;

box-shadow:inset 3px 3px 10px #06c;

}

不过box-shadow的内阴影使用在图片“img”元素上是没有任何效果的，例如：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>图片内阴影效果</title>

<style>

body{

background: #CCC;

}

img{

margin: 20px;

box-shadow:inset 5px 5px 10px #06c;

}

</style>

</head>

<body>

<img src="1.png" width="200"/>

</body>

</html>

上例的效果再次证实了box-shadow的inset内阴影直接运用在img上没有任何效果，但在实际Web项目中难免在图片上添加内阴影的效果。记得将border-radius运用在img上时，Webkit内核浏览器也无效果，最后在img外添加一个容器标签，并将img转化成外容器的背景图片，将border-radius运用在外容器上才有圆角效果的。借助这个思路，也在img标签外添加一个容器，例如div标签，这里不将img转化成div标签的背景，只是将box-shadow的内阴影使用在div标签上，例如：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>图片内阴影效果</title>

<style>

body{

background: #CCC;

}

.box-shadow{

margin: 20px;

display: inline-block;

box-shadow:inset 5px 5px 10px #06c;

}

img{

position: relative;

z-index: -1;

vertical-align: top;

}

</style>

</head>

<body>

<div class="box-shadow">

<img src="1.png" width="200"/>

</div>

</body>

</html>

4.多层阴影

前面几种都是单阴影效果的使用，其实box-shadow可以多层阴影同时使用，每层阴影之间使用“,”隔开。每层阴影的使用方法都和前面一样，例如：

.box-shadow{

width:200px;

height:100px;

border-radius: 5px;

margin: 30px;

border: solid 3px #CCC;

box-shadow: -5px 0px 5px red,0 5px 5px blue,5px 0 5px green,0 -5px 5px orange;

}

制作多层阴影效果时，不设置模糊半径，只设置扩展半径，并配合多个阴影颜色，还可以制作多色边框效果，代替border-color属性制作多色边框效果，例如：

.box-shadow{

width:200px;

height:100px;

border-radius: 5px;

margin: 30px;

border: solid 3px #CCC;

box-shadow:

0 0 0 1px red,

0 0 0 5px blue,

0 0 0 8px green,

0 0 0 12px yellow,

0 0 0 16px orange,

0 0 0 20px #06c,

0 0 0 24px lime;

}

使用box-shadow制作多色边框效果，需要注意模仿border的宽度。前面介绍过box-shadow的工作模式，计算宽度时需要减去前面阴影的值，才是显示的宽度。

在使用多层级box-shadow时，还需要特别注意阴影的顺序，最先写的阴影将显示在最顶层。