第5讲 DIV+CSS的边框与背景

2018年4月12日 C17f38

边框属性用于指定一个元素边框的样式、颜色、宽度、形状等等内容，背景属性用于指定一个元素的背景颜色、图像以及背景图像的相关设置。

## 1、边框border

### 1.1、边框样式border-style

边框样式属性指定要显示什么样外观的边界。它的属性值与含义如下：

none: 无边框，默认值

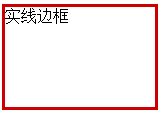
dotted: dotted:定义一个点线边框



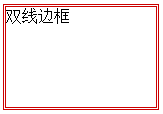
dashed: 定义一个虚线边框



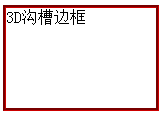
solid: 定义实线边框



double: 定义两个边框，两个边框的宽度和 border-width 的值相同



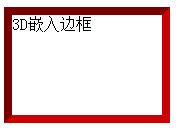
groove: 定义3D沟槽边框。效果取决于边框的颜色值



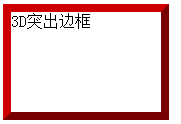
ridge: 定义3D脊边框。效果取决于边框的颜色值



inset:定义一个3D的嵌入边框。效果取决于边框的颜色值



outset: 定义一个3D突出边框。 效果取决于边框的颜色值



### 1.2、边框宽度border-width

通过 border-width 属性可以为边框指定宽度，边框的宽度实际是指边框线的粗细。

为边框指定宽度有两种方法：可以指定一个具体值，比如 2px 或 0.1em(单位为 px, pt堤, cm, em等)，也可使用 3 个关键字之一，它们分别是 thick 、medium（默认值） 和 thin。

例：

.one

{

border-style:solid;

border-width:5px;

}

.two

{

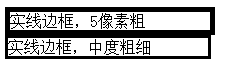
border-style:solid;

border-width:medium;

}

<div class=”one”>实线边框，5像素粗</div>

<div class=”two”>实线边框，中度粗细</div>



### 1.3边框颜色border-color

border-color属性用于设置边框的颜色。设置的颜色可以使用十六进制颜色值，也可以使用英文颜色名称，也可以使用rgb()函数：

* name - 指定颜色的名称，如 "red"
* RGB - 指定 RGB 值, 如 "rgb(255,0,0)"
* Hex - 指定16进制值, 如 "#ff0000"

还可以设置边框的颜色为"transparent"（透明）。

注意： border-color单独使用是不起作用的，必须得先使用border-style来设置边框样式。

.one

{

border-style:solid;

border-color:red;

width:180px;

}

.two

{

border-style:solid;

border-color:#C00;

border-width:1px;

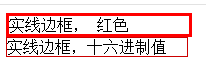
width:180px;

}

<div class=”one”>实线边框， 红色</div>

<div class=”two”>实线边框，十六进制值</div>

效果图如下：



### 1.4、单独设置边框

在CSS中，可以指定不同的边框使用不同的外观：

例

#ps

{

border-top-style:double;

border-right-style:solid;

border-bottom-style:dotted;

border-left-style:inset;

width:200px;

}

<div id=”ps”>不同的边框单独设置</div>



值得注意的是四条边框的衔接点的形式，见下例：

.bs

{

border-left:50px solid #C00;

border-right:50px solid #0C0;

border-top:50px solid #F3C;

border-bottom:50px solid #63C;

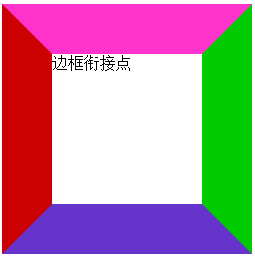
width:150px;

height:150px;

}

<div class="bs">边框衔接点</div>

效果如下图：



这样的45度角衔接方式，是很有特效价值的。当盒子的宽度与高度的尺寸，与边框的尺寸形成一定的比例关系时，特殊效果就会出现。

见下例：

.bs

{

border-left:50px solid #C00;

border-right:50px solid #0C0;

border-top:50px solid #F3C;

border-bottom:50px solid #63C;

width:0px;

height:0px;

}

<div class="bs"></div>



### 1.5、边框-简写属性

上面的例子用了很多属性来设置边框。这样代码会显得很繁冗。可以在一个综合属性中设置边框。例如

border-width：2px;

border-style：solid;

border-color:red;

可简写为

border:5px solid red;



## 2、边框特效

CSS3还允许用户创建圆角边框，向矩形添加阴影，使用图片来绘制边框等特效。

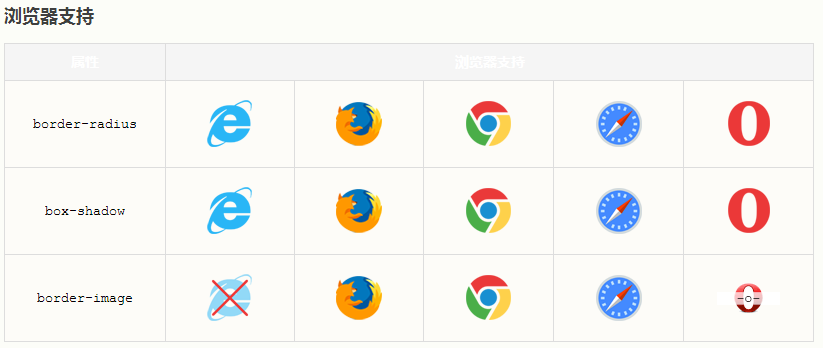
分别使用以下边框属性：

border-radius

box-shadow

border-image

对于这些属性，需要浏览器支持。具体情况如下：



### 2.1、圆角边框border-radius

在 CSS3 中，使用border-radius 属性来创建圆角：

border-radius 属性是一个简写属性，用于设置矩形盒子四个角的弧度属性。

默认值： 0

继承性： no

版本： CSS3

JavaScript 语法： object.style.borderRadius="5px"

语法

border-radius: 左上 右上 右下 左下;

说明：按左上、右上、右下、左下的顺序设置每个 radius 的四个值。值可以是长度值，也可以是%值。

**例子 1**

border-radius:2em;

等价于：

border-top-left-radius:2em;

border-top-right-radius:2em;

border-bottom-right-radius:2em;

border-bottom-left-radius:2em;



**例子 2**

border-radius: 2em 1em 4em 3em;

等价于：

border-top-left-radius: 2em;

border-top-right-radius: 1em;

border-bottom-right-radius: 4em;

border-bottom-left-radius: 3em;



实例

向 div 元素添加圆角：

div

{

border:2px solid;

border-radius:25px;

-moz-border-radius:25px;

Border:1px solid red;

width:250px;

height:100px;

}

<div>圆角div</div>



### 2.2、边框阴影box-shadow

在 CSS3 中，box-shadow 用于向方框添加阴影：

定义和用法

默认值： none

继承性： no

版本： CSS3

JavaScript 语法： object.style.boxShadow="10px 10px 5px #888888"

语法

box-shadow: h-shadow v-shadow blur spread color inset;

注释：box-shadow 向框添加一个或多个阴影。该属性是由逗号分隔的阴影列表，每个阴影由 2-4 个长度值、可选的颜色值以及可选的 inset 关键词来规定。省略长度的值是 0。

h-shadow 必需。水平阴影的位置。允许负值。

v-shadow 必需。垂直阴影的位置。允许负值。

blur 可选。模糊距离。

spread 可选。阴影的尺寸。

color 可选。阴影的颜色。

inset 可选。将默认的外部阴影 (outset) 改为内部阴影。

例1：

div

{

Border:1px solid red;

width:250px;

height:100px;

box-shadow: 10px 10px 5px #888888;

}

<div>边框阴影特效</div>



例2：

div

{

Border:1px solid red;

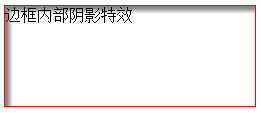
width:250px;

height:100px;

box-shadow: 5px 5px 5px #888888 inset;

}

<div>边框阴影特效</div>



### 2.3、边框图片border-image

通过 CSS3 的 border-image 属性，您可以使用图片来创建边框：

border-image 属性允许指定作为边框的图片。

语法：

border-image: source slice width outset repeat;

source: 用于指定要用于绘制边框的图像的路径

slice: 图像边界向内偏移，用不带单位的数值或百分比

width: 图像边界的宽度，使用像素为单位的数值

outset:图像超出边框的量，用不带单位的数值或百分比

repeat: 设置图像边界是否应重复（repeat）、拉伸（stretch）或铺满（round）

选看一个简单例子，使用图片创建围绕 div 元素的边框：

#a {

width:200px;

height:150px;

border:1px solid #666;

}

#b {

border-image:url(flower.jpg);

border-image-slice:3;

border-image-width:5px;

border-image-repeat:repeat;

width:200px;

height:150px;

}

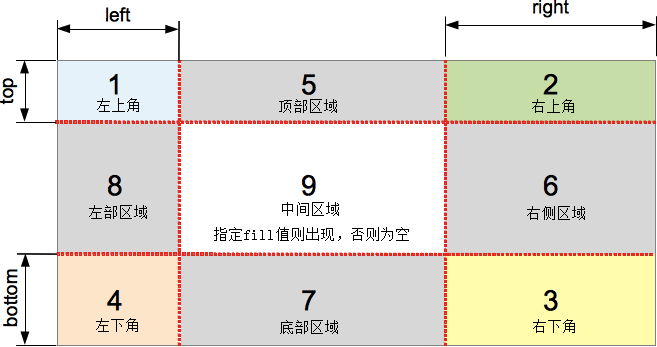
<div id="a">div的边框</div>

<div id="b">div的图片边框</div>



#### 2.3.1、边框切片属性border-image-slice

Border-image-slice属性也称为边框切片属性，它将作为边框的图像，按九宫格的形式进行分切。将边框图片切分成九片，如下图所示：



以一个81px的正方形位图，9个菱形图案，每个菱形图案为27\*27px 为例 ：



左上角、右上角、右下角、左下角为4个橙色菱形，顶部区域、右侧区域、底部区域、左侧区域为4个土黄菱形，中间块（贱称第九区）为透明块。

假如给定这个属性值：border-image-slice:27 27 27 27 那么它代表的意思是距离顶部内偏移区域27px处横切一刀，接着距离右侧内偏移区域27px竖切一刀，紧接着距离底部内偏移区域27px横切一刀，接着距离左侧内偏移区域27px竖切一刀，最后边框图片的切片border-image-slice:27 27 27 27的效果图如下：



分切以后，被分切的图像只能在边框宽度（border-width）内活动。比如盒子边框为border-width:54px，而分切图片为border-image-slice:27，因为图像4个顶角的宽度和高度都只有27px，但是盒子的边框是54px，因此图片就要被水平方向和垂直方向拉伸到切好跟盒子边框宽度等同，即27px的图像拉伸到54px停止。

**例1：当盒子边框宽度比被切图片边框大**

<style type="text/css">

#a {

border:54px solid #000;

border-image-source:url(border1.png);

border-image-slice:27 27 27 27;

width:81px;

height:81px;

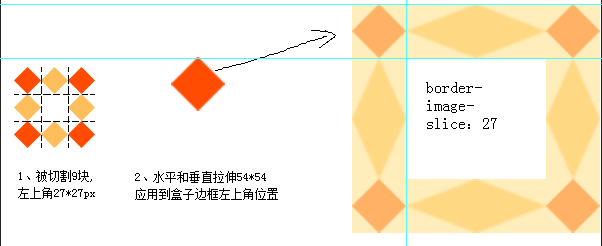
</style>

<div id="a">div的图片边框</div><div> 边框图片</div>

上例中，边框的宽度是54px，边框分切图片是27px，效果如下图：



CSS的拉伸机制是这样的：



**例2：盒子边框宽度比被切图片边框小**

#a {

border:14px solid #000;

border-image-source:url(border1.png);

border-image-slice:27 27 27 27;

width:81px;

height:81px;

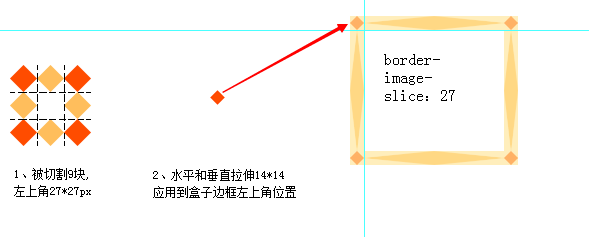
</style>

<div id="a">div的图片边框</div>

上例中，边框宽度为14px，边框图像的切片宽度依然为27PX，效果图如下：



其拉伸机制如下图：



如果设置了fill属性值，那么中间的空白区域就也将被图片填充,为了明白这个属性，我们另外使用一张边框图片来进行设置



**例3：fill属性**

#a {

border:14px solid #000;

border-image-source:url(border2.fw.png);

border-image-slice:27 27 27 27 ;

width:81px;

height:81px;

}

#b {

border:14px solid #000;

border-image-source:url(border2.fw.png);

border-image-slice:27 27 27 27 fill;

width:81px;

height:81px;

}

<div id="a">div的图片边框</div>

<div id="b">div的图片边框-带fill</div>

效果图如下：



#### 2.3.2、边框图片宽度border-image-width

border-image-width的作用代替盒子本身的边框宽度border-width。假如指定该属性，那么边框图片宽度就由border-image-width决定，假如不指定，那么图片边框宽度就由盒子的边框宽度border-width来固定。这个属性可以用具体像素、数字（表示几倍）和百分比来表示。

例如：

#b {

border:14px solid #000;

border-image-source:url(border2.fw.png);

border-image-slice:27 27 27 27;

border-image-width:27px 1 10% 27px;

width:81px;

height:81px;

}

<div id="b">div的图片边框</div>

边框图片的宽度分别为：上(27px)右（1倍）下（10%）左（27px）效果如下：



#### 2.3.3、边框图片外凸border-image-outset

border-image-outset的作用就让边框图片延伸到盒子外。有两种属性值length和number，前者是具体的像素单位，后者是数字，例：

#b {

border:27px solid #000;

border-image-source:url(border2.fw.png);

border-image-slice:27 27 27 27;

border-image-outset:27px 27px 27px 27px;

width:81px;

height:81px;

}

#a {

border:27px solid #000;

width:81px;

height:81px;

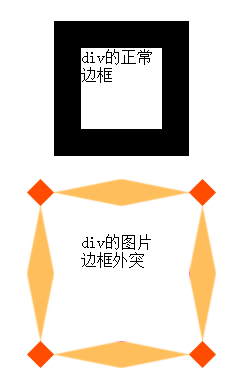
}

<div id="a">div的正常边框</div>

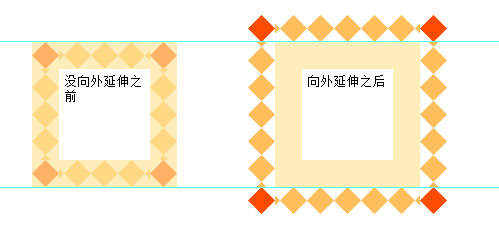
<p><br />

<div id="b">div的图片边框外突</div>

效果如下图：



其外突原理如下图：



#### 2.3.4、边框图片重复border-image-repeat

border-image-repeat的作用是设置边框背景是否重复，默认值为stretch，是拉伸的意思，指除了中间以外的其它8个区域做水平和垂直方向的拉伸，来填补边框的间隙；repeat（重复）是就是让8区域做完水平和垂直方向的复制图像，把边框之间的空隙填满；而round（环绕）是把8个区域分成均等区域，然后用背景图片切好能铺满整个边框空隙，不能多也不能少，正好合适。

例：

#a {

border:27px solid #000;

border-image-source:url(border2.fw.png);

border-image-slice:27 27 27 27;

border-image-repeat:stretch;

width:81px;

height:81px;

}

#b {

border:27px solid #000;

border-image-source:url(border2.fw.png);

border-image-slice:29 29 29 29;

border-image-repeat:repeat;

width:81px;

height:81px;

}

#c {

border:27px solid #000;

border-image-source:url(border2.fw.png);

border-image-slice:29 29 29 29;

border-image-repeat:round;

width:81px;

height:81px;

}

<div id="a">div的图片边框拉伸</div>

<div id="b">div的图片边框重复</div>

<div id="c">div的图片边框环绕</div>

效果如下图：

## 3、背景

CSS包含有多个背景属性，它们提供了对背景更强大的控制，这些属性中，主要的有以下几个：

1. background-color（背景颜色）
2. background-image（背景图像）
3. background-size（背景尺寸）
4. background-position（背景位置）
5. background-repeat（背景重复）
6. background-attachment（背景依附）
7. background-origin（背景起点）
8. background-clip（背景剪裁）

其中，有几个属性是CSS3才出现的，需要浏览器条件支持，如下图：



### 3.1、background-color 属性

background-color 属性定义了元素的背景颜色.

例1：定义页面的背景颜色:

body {background-color:#b0c4de;}

CSS中，颜色值可以有以下几种定义方式:

十六进制 - 如："#ff0000"

RGB - 如："rgb(255,0,0)"

颜色名称 - 如："red"

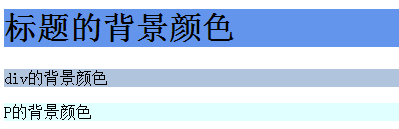
以下实例中,

例2：h1, p, 和 div 元素拥有不同的背景颜色:

h1 {background-color:#6495ed;}

p {background-color:#e0ffff;}

div {background-color:#b0c4de;}



### 3.2、background-image 属性

通过background-image属性添加背景图片。如果只有一张背景图像，默认情况下，背景图像进行平铺重复显示，以覆盖整个元素实体。如果有多张不同的背景图像，则图像之间用逗号隔开，所有的图片中显示在最顶端的为第一张。

例1把下面的两张图片作为DIV的背景图像:

#ss{

background-image:url(flower.png) ,url(bg1.jpg);

width:300px;

height:200px;

}

<DIV id="ss">图像背景示范</DIV>

效果如图所示：



可以分别给不同的图片设置多个不同的属性.

例2:

#ss{

background:url(flower.png) right bottom no-repeat,

url(bg1.jpg);

width:240px;

height:200px;

}

<DIV id="ss">图像背景示范</DIV>

效果如图所示：



注：一个元素，同时设置了背景图像与背景颜色的情况下，显示背景图像。

background属性是一个综合属性，它包含了了所有的背景属性，具体包括：

1. background-color
2. background-image
3. background-position
4. background-size
5. background-repeat
6. background-origin
7. background-clip
8. background-attachment

这些属性，可以全部设置，也可以忽略其中某些不设置。

### 3.3、background-size 属性

background-size 属性用于规定背景图片的尺寸。

默认值： auto

继承性： no

语法

background-size: length|percentage|cover|contain;

length：设置背景图像的高度和宽度。有两个值，第一个值设置宽度，第二个值设置高度。如果只设置一个值，则第二个值会被设置为 "auto"。

Percentage：以父元素的百分比来设置背景图像的宽度和高度，也有两个值，第一个值设置宽度，第二个值设置高度。如果只设置一个值，则第二个值会被设置为 "auto"。

Cover：把背景图像扩展至足够大，以使背景图像完全覆盖背景区域。如果尺寸设置过大，背景图像的某些部分也许无法显示在背景定位区域中。

Contain：把图像图像扩展至最大尺寸，以使其宽度和高度完全适应内容区域。

说明：在 CSS3 之前，背景图片的尺寸是由图片的实际尺寸决定的。在 CSS3 中，可以规定背景图片的尺寸，这就允许我们在不同的环境中重复使用背景图片。

可以用像素或百分比规定尺寸。如果以百分比规定尺寸，那么尺寸是相对于父元素的宽度和高度。

例1：调整背景图片的尺寸比盒子的尺寸小一些：

div

{

background:url(bg1.jpg);

background-size:150px 120px;

background-repeat:no-repeat;

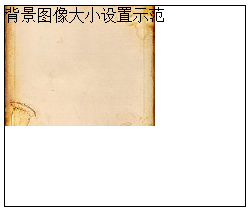
width:240px;

height:200px;

border:1px solid #000;

}

<DIV >背景图像大小设置示范</DIV>



例2：对背景图片进行拉伸，使其完成填充内容区域：

div

{

background:url(bg1.jpg);

background-size:40% 100%;

background-repeat:no-repeat;

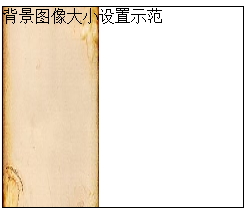
width:240px;

height:200px;

border:1px solid #000;

}

<DIV >背景图像大小设置示范</DIV>



### 3.4、background-postion属性

background-position 属性设置背景图像的起始位置。 这个属性设置背景原图像（由 background-image 定义）的位置，背景图像如果要重复，将从这一点开始。

语法格式：

background-position: x-value, y-value

x-value：x轴的起始点，可以是值，或百分比,也可以是位置的名称；

y-value:y轴的起始点，可以是值，或百分比,也可以是位置的名称；

如果两个参数只设置了一个，则第二个默认为“center”(居中)

例1：使用flower.png作为背景图像，让它显示在盒子的右上角

div

{

background-image:url(flower.png);

background-repeat:no-repeat;

background-position:right top;

width:240px;

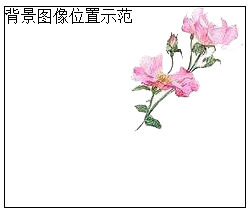
height:200px;

border:1px solid #000;

}

<DIV >背景图像位置示范</DIV>

效果如图所示：



例2：使用具体数值作为背景图像的起点

div

{

background-image:url(flower.png);

background-repeat:no-repeat;

background-position:80px 50px;

width:240px;

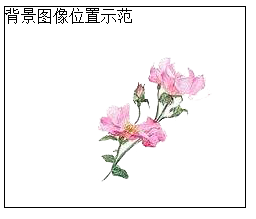
height:200px;

border:1px solid #000;

}

<DIV >背景图像位置示范</DIV>

效果如下图：



### 3.5、background-repeat属性

如果需要背景图像在重复方面作设置，可以使用 background-repeat 属性，它有6个属性值，含义分别如下：

no-repeat:不重复

repeat:重复（默认值）

repeat-x:在水平方向上重复

repeat-y:在垂直方向上重复

round:环绕(垂直方向拉伸，水平方向重复)

space:（以空白代替重复）

例1：使用花作为几个不同盒子的背景，并使用不同的重复属性

#a,#b,#c,#d,#e,#f

{

background-image:url(flower.png);

width:240px;

height:200px;

border:1px solid #000;

}

#a{background-repeat:no-repeat;}

#b{background-repeat:repeat;}

#c{ background-repeat:repeat-x;}

#d{background-repeat:repeat-y;}

#e{background-repeat:round;}

#f{background-repeat:space;}

<DIV id="a">背景图像不重复效果</DIV><br />

<DIV id="b">背景图像重复效果</DIV><br />

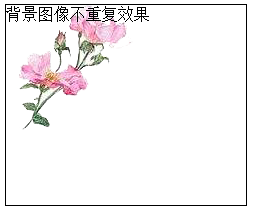
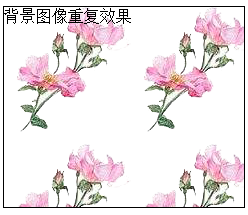
<DIV id="c">背景图像水平重复效果</DIV><br />

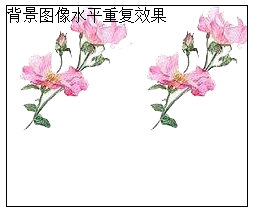
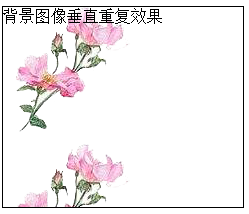
<DIV id="d">背景图像垂直重复效果</DIV><br />

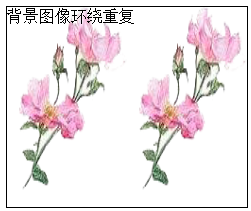
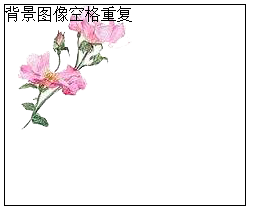
<DIV id="e">背景图像环绕重复</DIV><br />

<DIV id="f">背景图像空格重复</DIV><br />

运行效果：

### 3.6、background-attachment属性

background-attachment设置背景图像是否固定或者随着页面的其余部分滚动,它的值有两个：

* scroll：背景图片随页面的其余部分滚动。这是默认
* fixed：背景图像是固定的

### 3.7、background-origin 属性

background-origin（起点，原点）属性用于规定背景图片的定位区域。背景图片可以放置于 content-box、padding-box 或 border-box 区域。

background-origin 属性规定 background-position（位置）属性相对于什么位置来定位。

注意：如果背景图像的 background-attachment 属性为 "fixed"，则该属性没有效果。

默认值： padding-box

继承性：no

语法

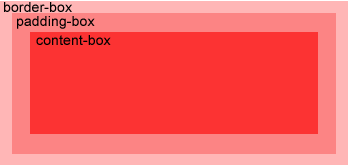
background-origin: padding-box|border-box|content-box;

padding-box：背景图像相对于内边距框来定位。

border-box：背景图像相对于边框盒来定位。

content-box：背景图像相对于内容框来定位。

其定位效果如图所示：



例1：在DIV中根据不同的定位显示背景图片：

#a,#b,#c

{

background-image:url(bg1.jpg);

background-size:100% 100%;

width:240px;

height:200px;

border:1px solid #000;

padding:20px 20px;

background-position:10px;

background-repeat:no-repeat;

}

#a{background-origin:border-box;}

#b{background-origin:content-box;}

#c{background-origin:padding-box;}

<DIV id="a">背景图像定位边框盒</DIV><br />

<DIV id="b">背景图像定位内容盒</DIV><br />

<DIV id="c">背景图像定位内边距</DIV><br />

效果分别如下图：

### 3.8、background-clip 属性

background-clip属性用于规定背景的绘制区域。

默认值： border-box

继承性： no

语法

background-clip: border-box|padding-box|content-box;

border-box：背景被裁剪到边框盒，相当于没剪裁

padding-box：背景被裁剪到内边距框。

content-box：背景被裁剪到内容框。

例1:使用不同的设置对背景进行剪裁

#a,#b,#c

{

background-image:url(bg1.jpg);

background-size:100% 100%;

width:240px;

height:200px;

border:1px solid #000;

padding:20px 20px;

background-repeat:no-repeat;

}

#a{background-clip:border-box;}

#b{background-clip:content-box;}

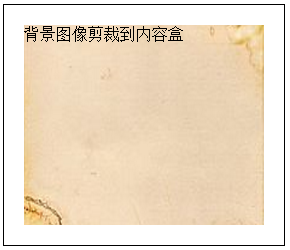
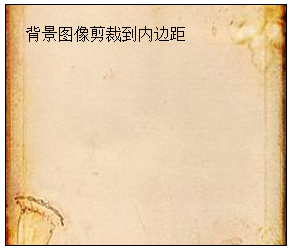
#c{background-clip:padding-box;}

<DIV id="a">背景图像剪裁到边框盒</DIV><br />

<DIV id="b">背景图像剪裁到内容盒</DIV><br />

<DIV id="c">背景图像剪裁到内边距</DIV><br />.

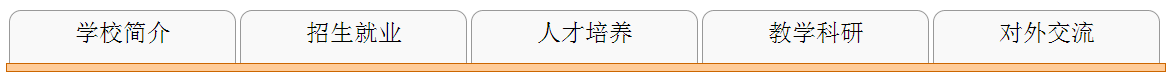
效果如图所示：

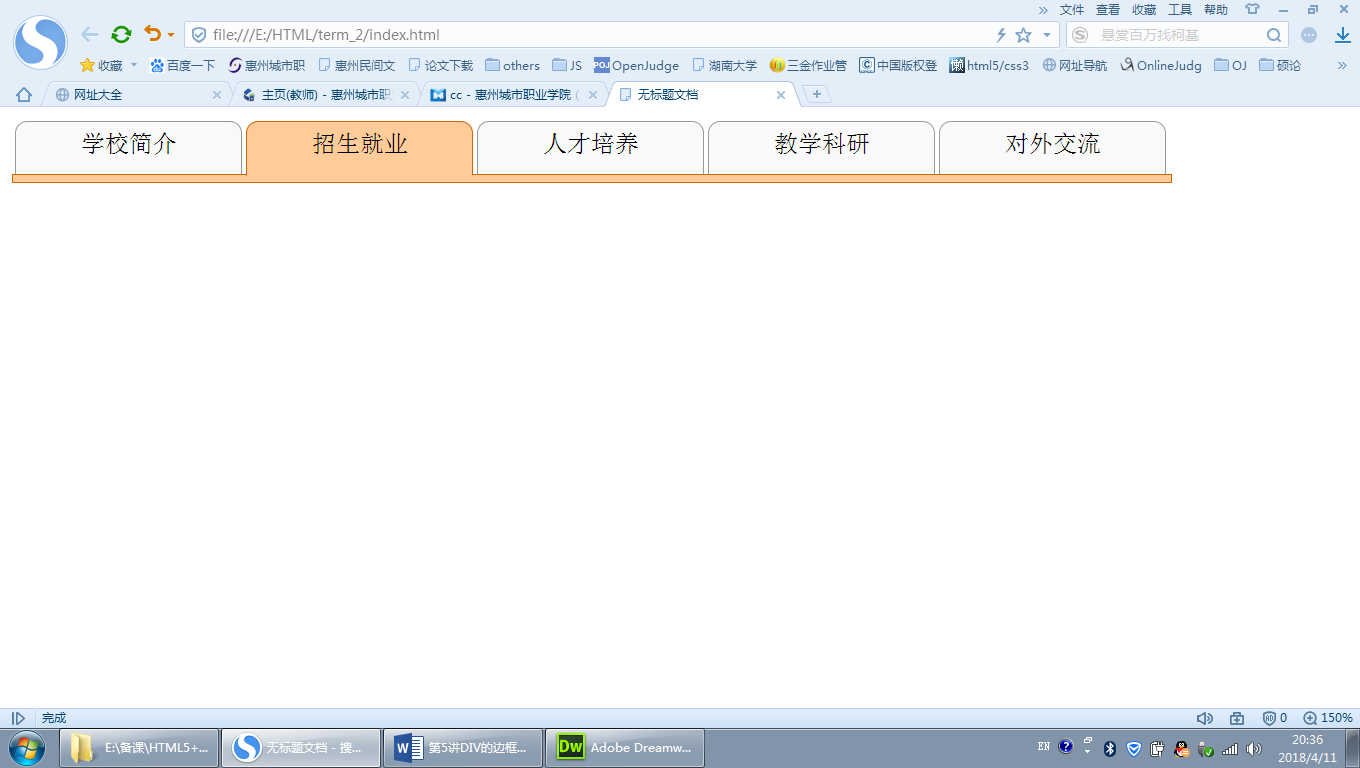
## 4、课堂练习：

1、请利用边框属性，结合其它必须的CSS属性，实现以下导航条效果：

导航项上部圆角，



当鼠标移到某项上面时，该项浮出，边框线与导航线成一体



2、利用CSS知识，实现以下位置效果：

