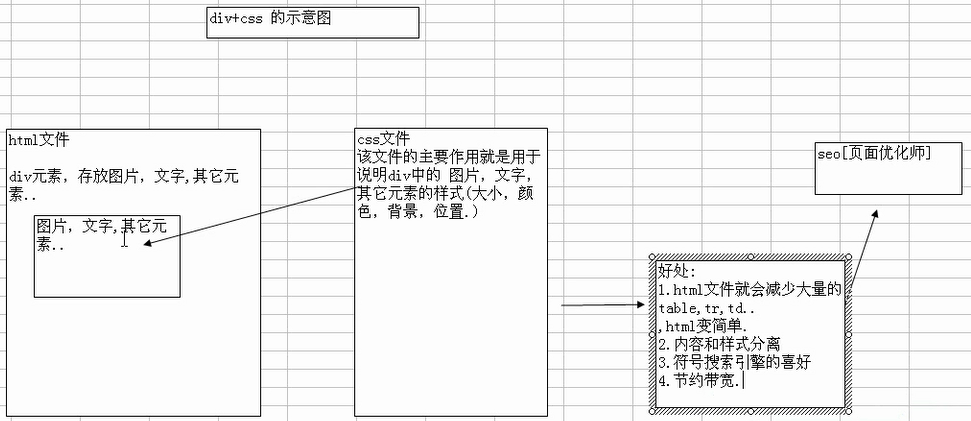
**div+css详解**

**一．div+css的介绍**

div是用于存放html元素，文字，图片，视频的元素

css 是层叠样式表，用于指定div中的内容的样式

原理图：



div+css的快速体验案例：

test.html：

<!DOCTYPE HTML PUBLIC "-//w3c//DTD HTML 4.0 Transitional//EN">

<html>

<head>

<title>快速入门</title>

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

</head>

<body>

<div class="style1">

<table>

<tr><td>1</td><td>2</td><td>3</td></tr>

<tr><td>1</td><td>2</td><td>3</td></tr>

<tr><td>1</td><td>2</td><td>3</td></tr>

</table>

</div>

</body>

</html>

my.css中指定：

.style1{

width:300px;

height:200px;

border:1px solid red;

margin:100px 0px 0px 200px;

}

.style1 table{

border:1px solid black;

width:298px;

height:190px;

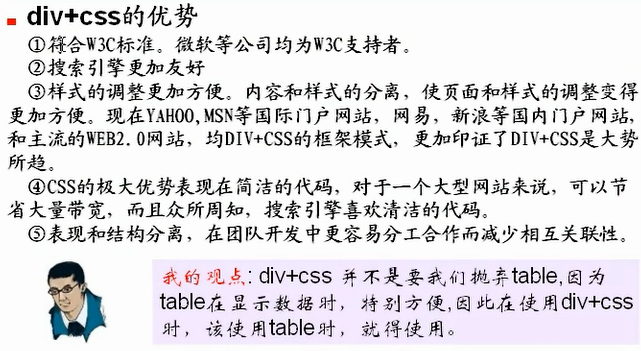
}

.style1 table td{

border: 1px solid black;

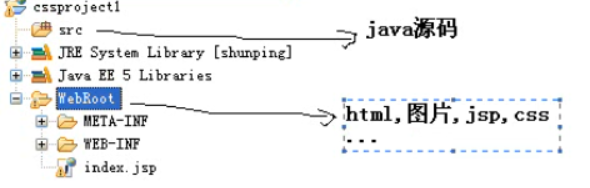
text-align:center;

}



使用<span>元素来编写

**使用myeclipse来开发，因为myeclipse有提示功能**



<span style="font-size:30px;color: blue;">栏目一</span><br/>

从使用span元素我们可以看到，css的基本语法

<元素名 style=”属性名：属性值；属性名：属性值2；”/>

元素可以是html的任意元素：

属性名：属性值要参考 w3c组织给出的参考文档

◆**使用css可以统一网站的风格**

css分类：外部css 内部css

<!-- DOCTYPE文档类型用于指定 DTD(说明当前 这个html的版本)-->

<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<title>css1.html</title>

<!-- 这个 keywords是给搜索引擎看-->

<meta http-equiv=*"keywords"* content=*"keyword1,keyword2,keyword3"*>

<meta http-equiv=*"description"* content=*"this is my page"*>

<!-- 告诉浏览器文件是什么编码 -->

<meta http-equiv=*"content-type"* content=*"text/html; charset=UTF-8"*>

<!-- 引入css文件 -->

<!--<link rel="stylesheet" type="text/css" href="./styles.css">-->

<style type=*"text/css"*>

.style1{

font-size: *20*;

color: *green*;

font-style: *normal*;

font-weight: *bold*;

}

</style>

</head>

<body>

<span class=*"style1"*>栏目一</span></br>

<span class=*"style1"*>栏目一</span>

</body>

</html>

**使用滤镜**

<style type=*"text/css"*>

/\*使用滤镜将图片变成黑白色\*/

img {

-webkit-filter: *grayscale(100%)*;

}

</style>

/\*使用滤镜\*/

**a***:link* **img** {

-webkit-filter: *grayscale(100%)*;

}

**a***:hover* **img**{

-webkit-filter: *grayscale(0%)*;

}</style>

<!--<link rel="stylesheet" type="text/css" href="./styles.css">-->

</head>

<body>

<a href=*"#"*><img src=*"images/1.jpg"*width=*170px* border=*1*/></a>

<a href=*"#"*><img src=*"images/2.jpg"*width=*170*px border=*1*/></a>

</body>

**二.css的4种选择器**

**类选择器 id选择器 html元素选择器 通配符选择器**

**1．类选择器的基本语法:**

.类选择器名{

属性名:属性值;

…

}

my.css文件

.style1{

font-size: *20px*;

font-weight: *bold*;

background-color: *purple*;

}

select1.html文件

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

</head>

<body>

<span class="style1">栏目一</span>

<span class="style1">栏目二</span>

<span class="style1">栏目三</span>

<span class="style1">栏目四</span></body>

**2．id选择器 基本语法：**

#id选择器名{

属性名：属性值；

}

案例：

#style2{

font-size: *30px*;

background-color: *skyblue*;

}

在html文件中如果要引用id选择器，则

<元素 id=”id选择器的名称”></元素>

<span id="style2">这是一则重要的新闻</span>

**3．Html选择器**

/\*html选择器body(button,input,form...)\*/

body{

color:*silver*;

}

**结论：当一个元素同时被id选择器 类选择器 html选择器修饰时，优先级为：**

**id选择器>类选择器>html选择器>通配符选择器**

案例：假设，我们希望所有的超链接

默认样式是黑色，24px，没有下划线

当鼠标移动到超链接时，自动出现下划线

点击后，超链接变成红色。

my.css文件

a:link{

color:*black*;

font-size:*24px*;

text-decoration:*none*;

}

a:hover{

text-decoration:*underline*;

}

a:visited{

color:*red*;

}

html文件：

<a href=*"http://www.baidu.com"*target=*"\_blank"*>百度首页</a><br/>

<a href=*"http://www.sohu.com"*target=*"\_blank"*>搜狐首页</a>

**4．通配符选择器**

如果希望所有的元素都符合某一种样式，可以使用通配符选择器。

基本语法：

/\*使用通配符选择器对外边距和内边距清零\*/

\*{

margin: *0*;

padding: *0*;

}

\*{

/\*margin: 0; 将外边距清零\*/

/\*margin-top:10px;分别设置四个方向的外边距

margin-left:10px;

margin-right:0px;

margin-bottom:opx;\*/

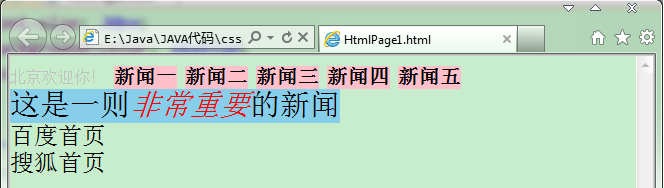
/\*margin:10px 0px 0px 10px;顺时针方向：上，右，下，左 \*/

margin:*10px* *0px* *0px*;/\*10表示上，左右，下\*/

padding: *0*;/\*将内边距清零，padding的规范跟margin一样/

}

**5．父子选择器**



针对：

<span id=*"style2"*>这是一则<span>非常重要</span>的新闻</span><br/>

my.css添加一个父子选择器

/\*父子选择器\*/

#style2 span{

font-style:*italic*;

color:*red*;

}

注意（1）子选择器标签必须是html可以识别的标记

（2）父子选择器可以有多级

#style2 span span{

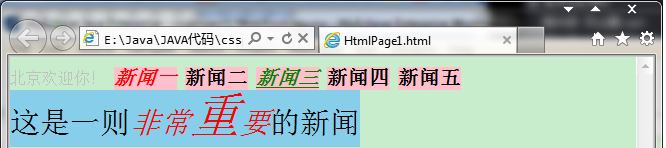
font-size: *50px*;

}

（3）父子选择器可以**适用于id选择器和class选择器**

（4）如果一个元素被id选择器和class选择器同时修饰时，id选择器的优先级>class选择器

（5）**一个元素最多有一个id选择器，但是可以有多个类选择器**

例 

需求：希望新闻三 下划线，同时斜体

方法1：可以给新闻三配置一个id选择器

方法2：再指定一个类选择器

my.css中添加

.style1{

font-size: *20px*;

font-weight: *bold*;

background-color: *green*;

}

.style4{

font-style: *italic*;

text-decoration: *underline*;

}

Html文件中如何使用多个class选择器：

<span class="style1 style4">新闻三</span>

**注意：**

**在引用多个class选择器的时候，用空格隔开**

**当class选择器发生冲突时，在css文件中，以最后出现的class选择器样式为准**

（6）**在css文件中，如果有多个类/id选择器，它们都有相同的部分，我们可以把相同的css样式放在一起，看案例：**

案例：my.css

/\*招生广告\*/

.ad\_stu{

width :136px;

height:196px;

background-color:#FC7E8C;

margin:5px 0 0 5px;

float:left

}

/\*广告2\*/

.ad\_2{

width :457px;

height:196px;

background-color:#7CF574;

margin:5px 0 0 6px;

float:left

}

/\*房地产广告\*/

.ad\_house{

width :152px;

height:196px;

background-color:#7CF574;

margin:5px 0 0 5px;

float:left

}

在有些css中，我们可以把多个class选择器，id选择器，html选择器，共同的部分提取出来，写在一起，这样的好处是可以简化css文件

我们可以把上面的css文件改写成：

/\*招生广告\*/

.ad\_stu{

width :136px;

background-color:#FC7E8C;

margin:5px 0 0 5px;

}

/\*广告2\*/

.ad\_2{

width :457px;

background-color:#7CF574;

margin:5px 0 0 6px;

}

/\*房地产广告\*/

.ad\_house{

height:196px;

background-color:#7CF574;

margin:5px 0 0 5px;

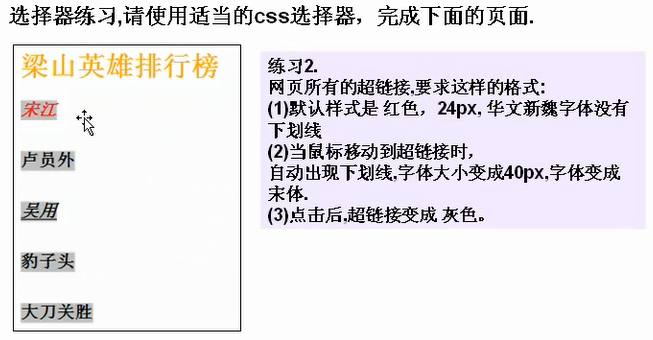
}

**.ad\_stu,.ad\_2,.ad\_house{**

**width :152px;**

**float:left**

**}**



见test2.html

**6．块元素和行内元素**

（1）**行内元素，又叫内联元素**

内联元素只能容纳文本或者其他内联元素，常见内联元素<span><a><input>

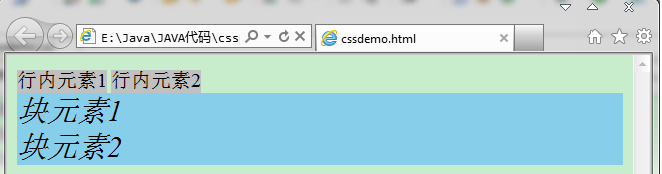
特点是：只占内容的宽度，默认不换行，行内元素一般放文本或者其它的行内元素。

（2）**块元素**

块元素一般都从新行开始，可以容纳文本，其它内联元素和其它块元素，即使内容不能占满一行，块元素也要把正行占满，常见块元素 <div><p>

特点：不管内容有多少，都要换行，同时占满整行，块元素可以放文本，行内元素，块元素。

（3）**一些css属性对行内元素不生效**，比如margin，left，right，width，height，建议尽可能使用块元素定位。（与浏览器类版本和类型有关）



<span class=*"s1"*>行内元素1</span>

<span class=*"s1"*>行内元素2</span>

<div class=*"s2"*>块元素1</div>

<div class=*"s2"*>块元素2</div>

.s1{

background-color: *silver*;

font-size:*20px*;

}

.s2{

background-color: *skyblue*;

font-size:*30px*;

font-style: *italic*;

}

（4）**块元素和行内元素的转换**

如果我们希望一个元素按照块元素的方式显示，则： display：block；

<span style=”display：block;” class=”s2”>abc</span>

也可以写到css中

如果我们希望一个元素按照行内元素的方式显示，则： display：inline；

<div style=”display：inline;” class=”s2”>hello</div>

**css核心内容-----流**

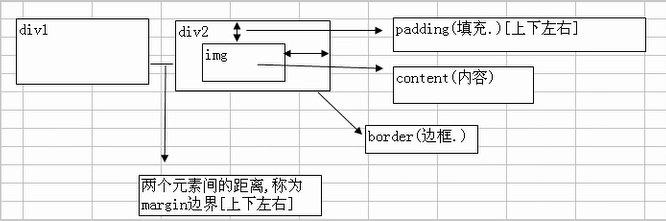
标准流/非标准流

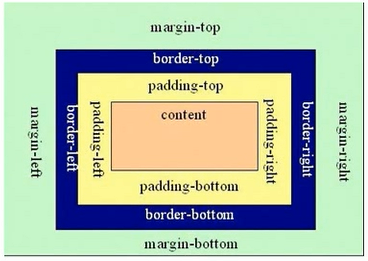
**标准流**：在网页布局中，写在前面的元素出现在前面，写在后面的元素，显示在后面，这是默认的布局方式，也称标准流。

**非标准流**：在实际网页布局中，我们可能需要使用非标准流的方式来布局（让某个元素脱离它本身的位置）。

**css盒子模型**

盒子模型的原理图





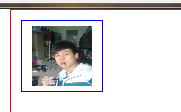
细节说明：

1）html元素都可以看成一个盒子

盒子模型的参照物不一样，则使用的css属性不一样，比如从div1的角度看，是margin-right，从div2的角度看，则是margin-left

3）如果不希望破坏整个外观，则尽量使用margin布局，因为padding可能会改变盒子的大小（相当于这个盒子有弹性），margin如果过大，盒子内容被挤到盒子外边去，但是盒子本身没有变化

案例:



Box1.html:

Html版本要改成下面的：

<!DOCTYPEHTMLPUBLIC"-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<div class=*"s1"*><img src=*"images/1.jpg"*></img></div>

Box1.css：

body{

border:*1px* *solid* *red*;/\*这里我们给body指定了宽度，样式，颜色（顺序可以随意）\*/

width: *800px*;

height:*1000px*;

margin:*0* *auto*;/\*0表示上下边距为0，auto左右自动居中\*/

}

.s1{

border:*1px* *solid* *blue*;

width:*80px*;

height:*70px*;

margin-top: *10px*;

margin-left: *10px*;

}

.s1 img{

width:*60px*;

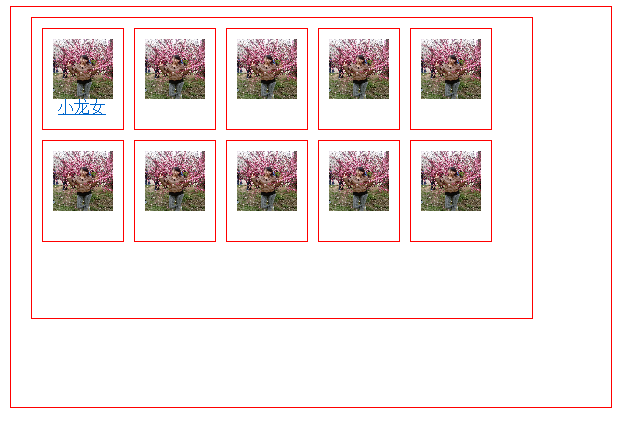
height:*60px*;

margin-left: *10px*;

margin-top: *5px*;

}

盒子模型的综合案例（可以当做一个模板来使用）



**Boxzong.html文件 float2.html也实现了这个页面 用不同的方法**

<body>

<div class=*"s1"*>

<ul class=*"s2"*>

<li ><img src=*"images/4.jpg"*></img><a href=*"#"*>小龙女</a></li>

<li ><img src=*"images/4.jpg"* ></img></li>

<li ><img src=*"images/4.jpg"* ></img></li>

<li ><img src=*"images/4.jpg"*></img></li>

<li ><img src=*"images/4.jpg"*></img></li>

<li ><img src=*"images/4.jpg"* ></img></li>

<li ><img src=*"images/4.jpg"* ></img></li>

<li ><img src=*"images/4.jpg"*></img></li>

<li ><img src=*"images/4.jpg"* ></img></li>

<li ><img src=*"images/4.jpg"*></img></li>

</ul>

</div>

</body>

**Boxzong.css文件**

body{

margin: *0*;

padding: *0*;

}

/\*最外面的\*/

.s1{

border: *red* *solid* *1px*;

width:*600px*;

height:*400px*;

margin-top: *20px*;

margin-left: *100px*;

}

/\*控制整个图片的区域\*/

.s2{

border: *red* *solid* *1px*;

width:*500px*;

height:*300px*;

margin-top: *10px*;

margin-left: *20px*;

list-style-type: *none*; /\*将点清除\*/

padding: *0*;

}

/\*控制单个图片区域\*/

.s2 li{

width:*80px*;

height:*100px*;

margin-top: *10px*;

margin-left: *10px*;

border: *red* *solid* *1px*;

float: *left*;/\*左浮动\*/

}

.s2 a{

margin-left: *15px*;

}

/\*控制图片的\*/

.s2 li img{

width:*60px*;

height:*60px*;

margin-top: *10px*;

margin-left: *10px*;

}

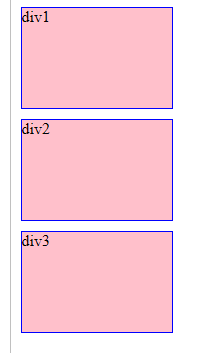
**练习：**



**浮动**

浮动是一个重要的概念：分为左浮动，右浮动，清除浮动。

浮动必要性，因为默认情况下，div纵向排列：



**Float1.html文件**

<div class=*"div1"*>div1</div>

<div class=*"div1"*>div2</div>

<div class=*"div1"*>div3</div>

**Float1.css文件**

body{

border:*1px* *solid* *silver*;

width: *800px*;

height:*1000px*;

margin:*0* *auto*;

}

.div1{

border:*1px* *solid* *blue*;

background-color: *pink*;

width:*150px*;

height:*100px*;

margin-top: *10px*;

margin-left: *10px*;

}

如果希望div1向右面显示，这时我们使用右浮动：



**Float1.html文件**

<div class=*"div1"* id=*"div2"*>div1</div>

<div class=*"div1"*>div2</div>

<div class=*"div1"*>div3</div>

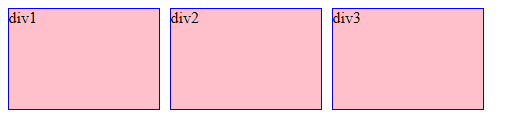
**Float1.css文件中只许添加一个id选择器**

#div2{

float:*right*; /\*使该元素尽量向右移动，直到碰到父元素的右边界\*/

}

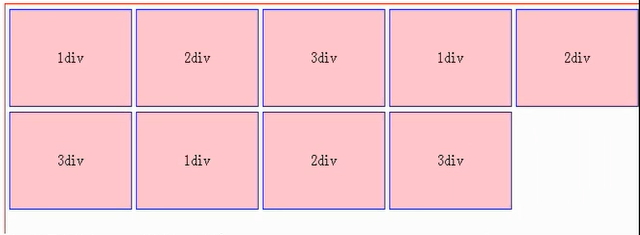
如果我们希望，所有的元素，横向排列，则使用到左浮动：



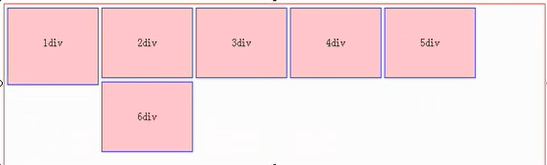
这时，对div1的css添加：

float：left；让出自己右边的空间使下一个元素使用

特别注意：如果一行宽度不够排下所有的div，则会自动换行：



**当然，如果有某个div的过大，则会卡住别的div**



特别强调：浮动对块元素和行内元素都是生效的。

**css定位**

css的定位有以下几种：4

**1.static定位**

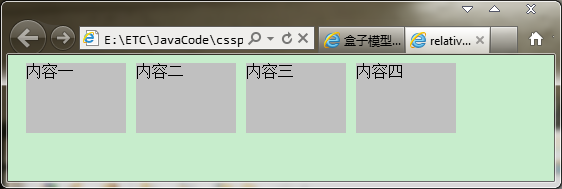
这个是默认的方式，对static而言，left和right是不生效的。

**2.relative定位**（相对定位）

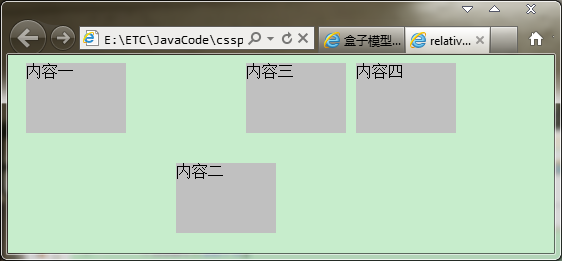
元素框偏离某个位置（left 和 top），元素仍然保持为定位前的形状，它原本所占的空间仍保留，从这一角度看，好像该元素仍然在文档流/标准流中一样。

特别说明：1.relative的参照点是它原来的位置，进行移动

案例：



改为：



代码：

<body>

<div class=*"style1"*>内容一</div>

<div id=*"special"* class=*"style1"*>内容二</div>

<div class=*"style1"*>内容三</div>

<div class=*"style1"*>内容四</div>

</body>

在css中：

.style1{

width:*100px*;

height:*70px*;

background-color:*silver*;

float:*left*;

margin-left:*10px*;

}

#special{

position:*relative*; /\*使用相对定位\*/

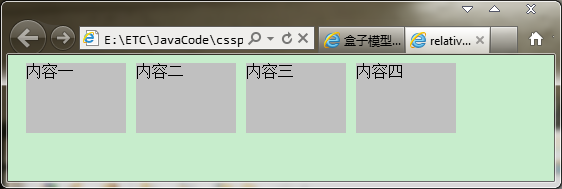
left:*40px*; /\*在原来的位置的基础上，向右移动的大小（如果希望向左移动，则值就是负数）\*/

top:*100px*; /\*在原来的位置的基础上，向下移动的大小（如果希望向上移动，则值就是负数）\*/

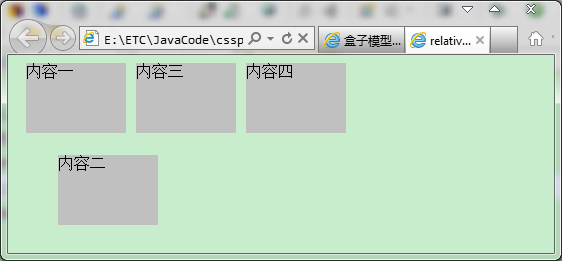
}

**3.absolute定位（绝对定位）**

相当于元素从原来的位置脱离，并让出自己的空间，后面的元素就会占有让出的空间。



改为：



代码：

<body>

<divclass=*"style1"*>内容一</div>

<div id=*"special"* class=*"style1"*>内容二</div>

<divclass=*"style1"*>内容三</div>

<divclass=*"style1"*>内容四</div>

</body>

在css中：

.style1{

width:*100px*;

height:*70px*;

background-color:*silver*;

float:*left*;

margin-left:*10px*;

}

#special{

position:*absolute*;/\*使用绝对定位\*/

left:*40px*; /\*向右移动的大小（如果希望向左移动，则值就是负数）\*/

top:*100px*; /\*向下移动的大小（如果希望向上移动，则值就是负数）\*/

}

特别说明：

这里特别强调一点，absolute定位是相对于离自己最近的那个非标准流盒子而言的。

**4.fixed定位**

元素框的表现类似于将position设置为absolute，不过其包含块是视窗本身。