

第三天 二列和三列布局

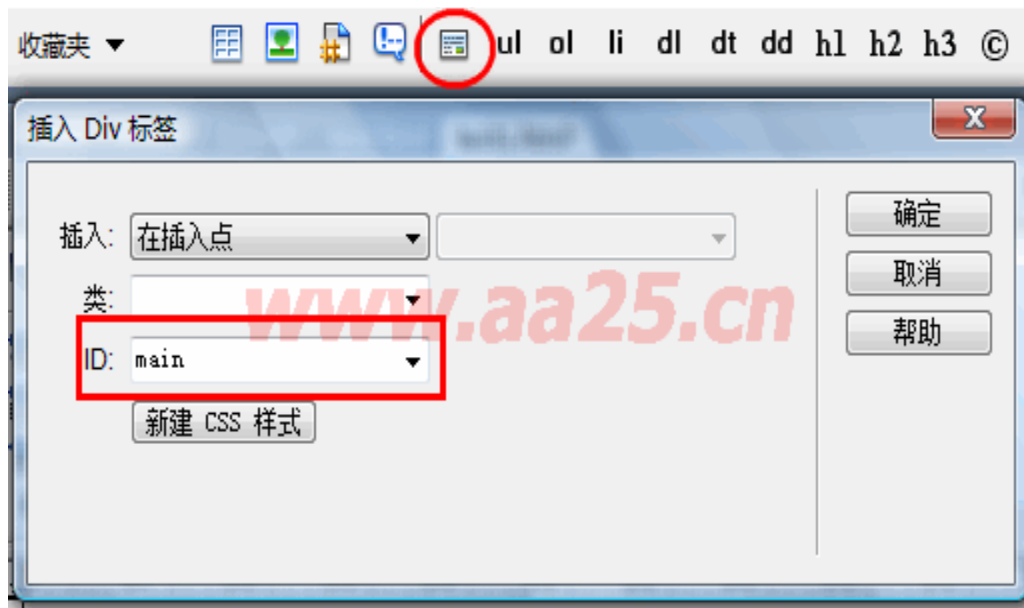
今天学习《十天学会 web 标准 (div+css)》的二列和三列布局，将涉及到以下内容 and 知识点

- 二列自适应宽度
- 二列固定宽度
- 二列固定宽度居中
- xhtml 的块级元素 (div) 和内联元素 (span)
- float 属性
- 三列自适应宽度
- 三列固定宽度
- 三列固定宽度居中
- IE6 的 3 像素 bug

一、两列自适应宽度

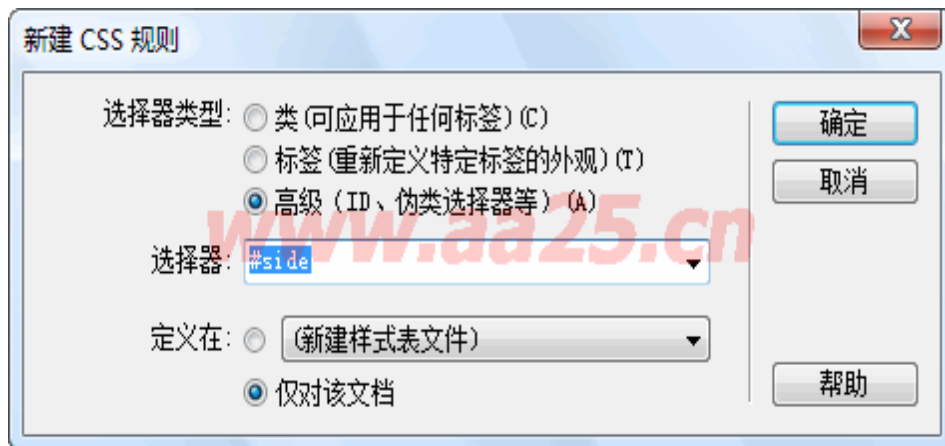
下面以常见的左列固定右列自适应为例，因为 div 为块状元素，默认情况下占据一行的空间，要想让下面的 div 跑到右侧，就需要借助 css 的浮动来实现。首先创建 html 代码如下：

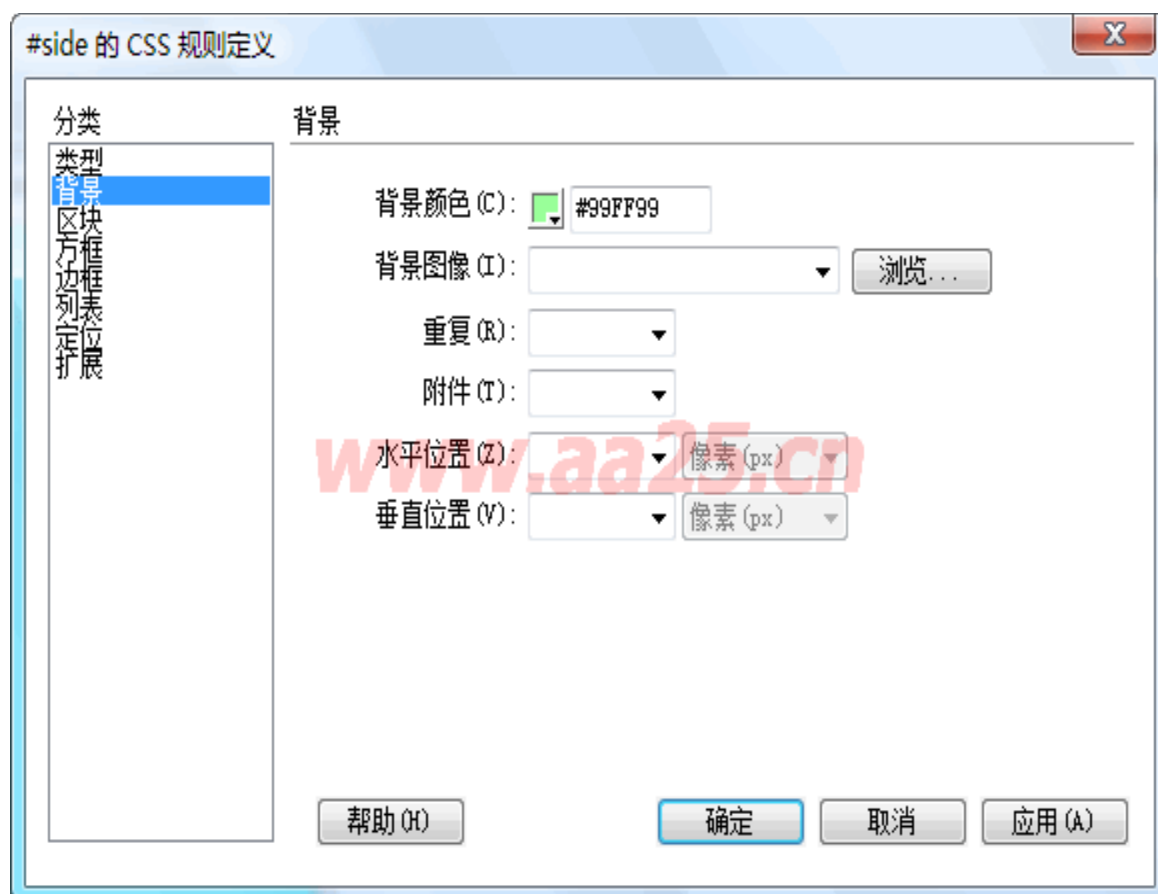
```
<div id="side">此处显示 id "side" 的内容</div>  
<div id="main">此处显示 id "main" 的内容</div>
```



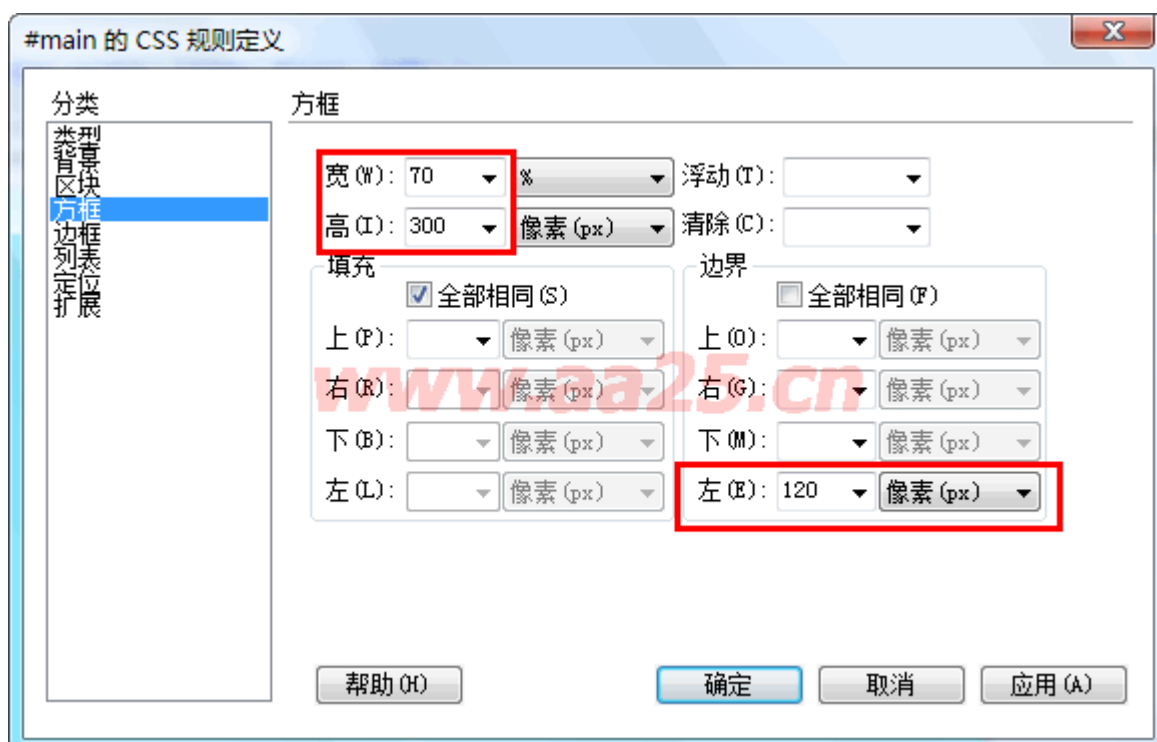
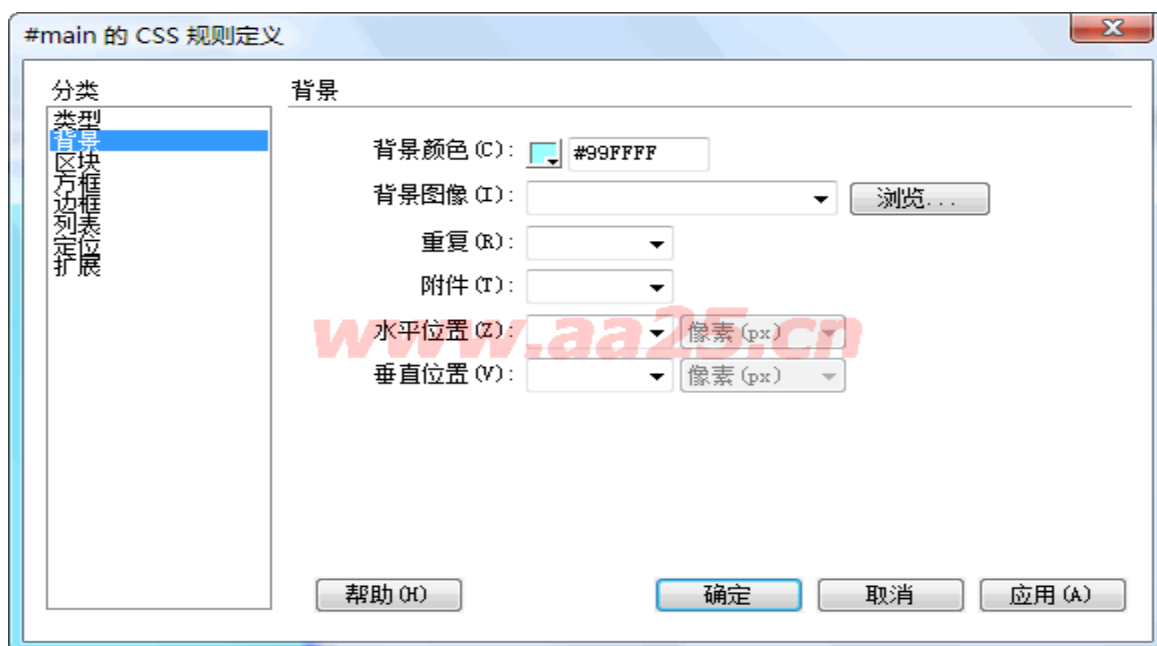
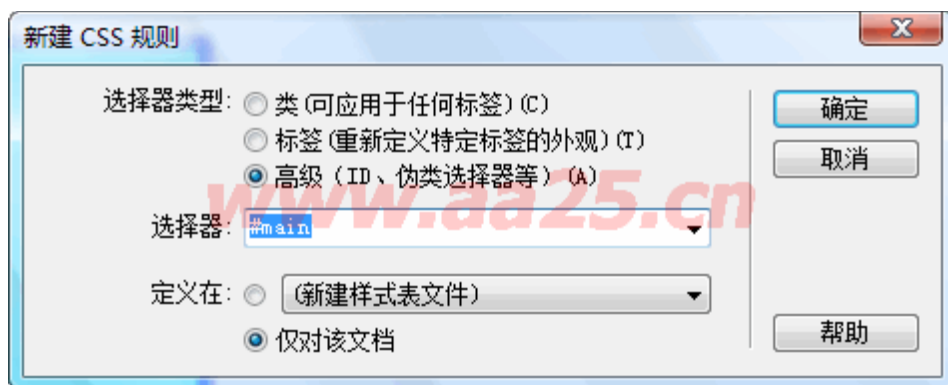
按照如图所示的创建方法，把两个 div 都创建出来，或手工写出代码。div 创建完成后，开始创建 css 样式表，代码如下：

```
#side { background: #99FF99; height: 300px; width: 120px; float: left; }  
#main { background: #99FFFF; height: 300px; width: 70%; margin-left: 120px; }
```



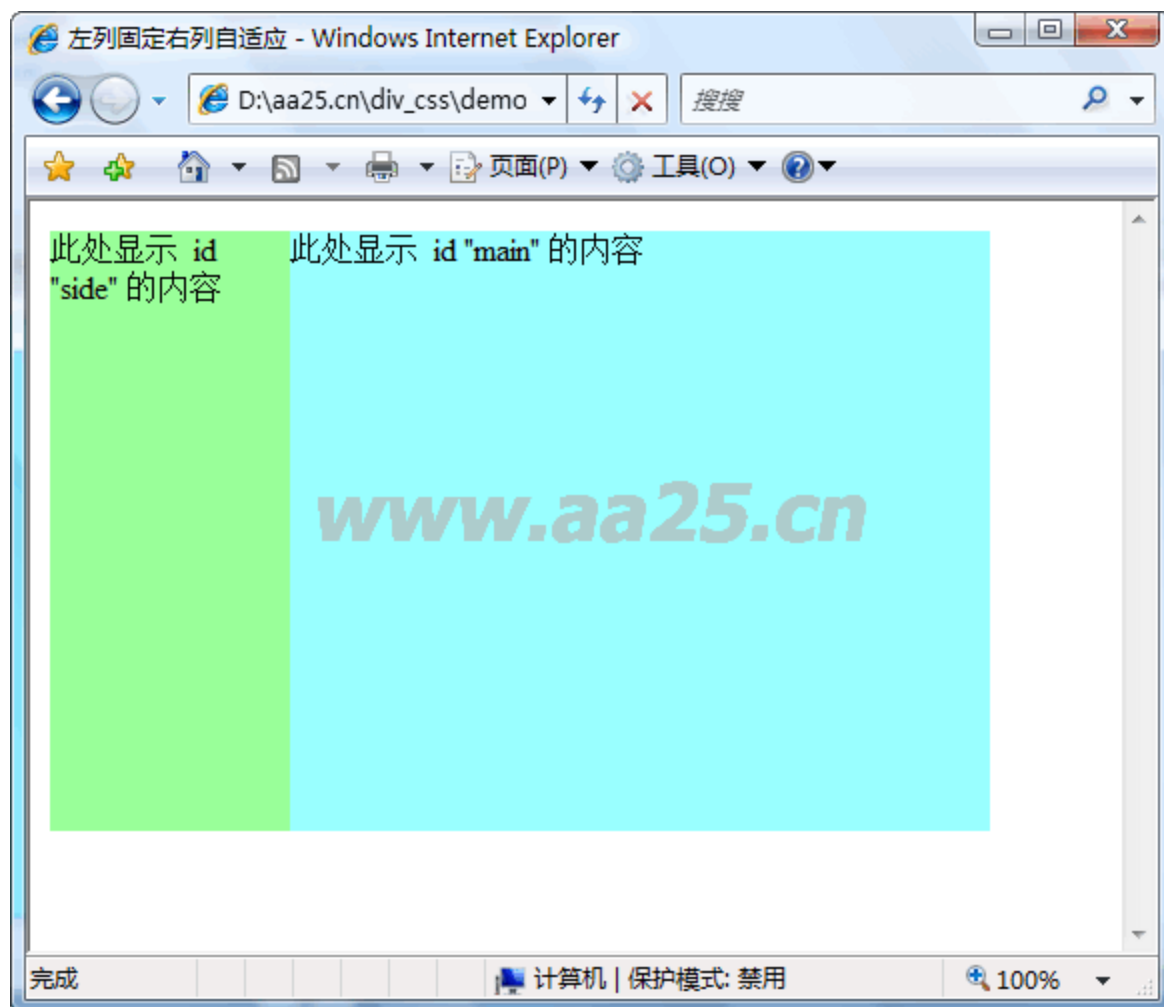


先创建#side 的样式，为了便于查看，设置了背景色。注意，side 的浮动设置为向左浮动；



然后创建#main 样式，注意这里设置#main 的左边距为 120px。预览结果如

下：



当我们拖动浏览器窗口变大变小时，#main 的宽度也会跟着改变。这里设置 margin-left:120px; 正好让出#side 占去的 120px 宽度，如果这里设置为 122px 的话，中间将出现 2px 的空隙，大家可以试一下。



```
<!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>

#side { background: #99FF99; height: 300px; width: 120px; float:
left; }

#main { background: #99FFFF; height: 300px; width: 70%; margin-left:
120px; }

</style>

</head>

<body>

<div id="side">此处显示 id "side" 的内容</div>

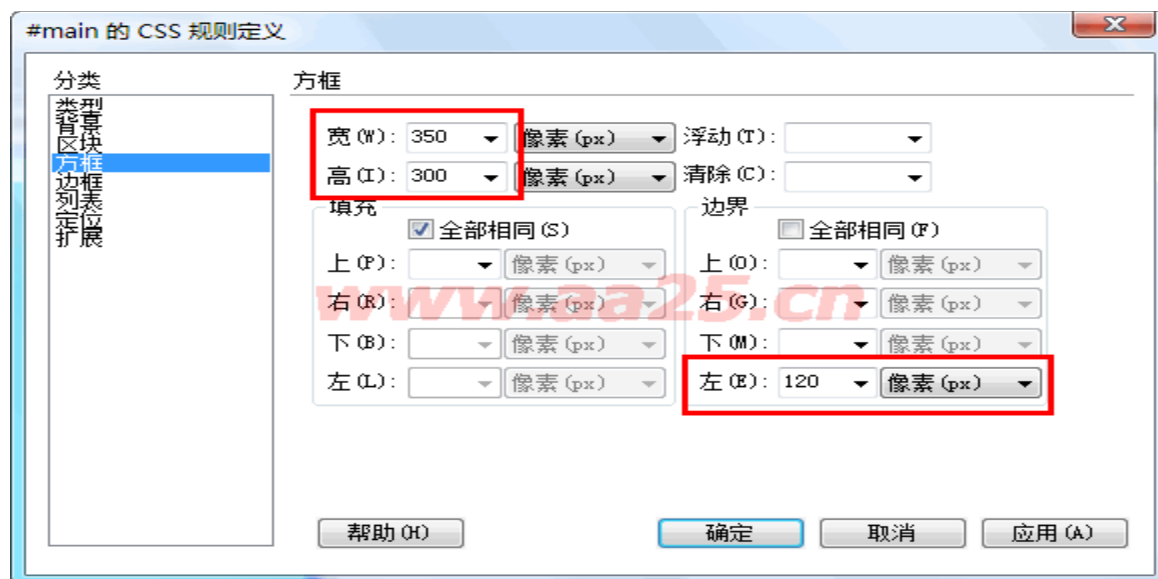
<div id="main">此处显示 id "main" 的内容</div>

</body>

</html>
```

二、两列固定宽度

有了前面的基础，两列固定宽度就容易多了，只需要把#main的宽度由百分比改为固定值，如下图：



三、两列固定宽度居中

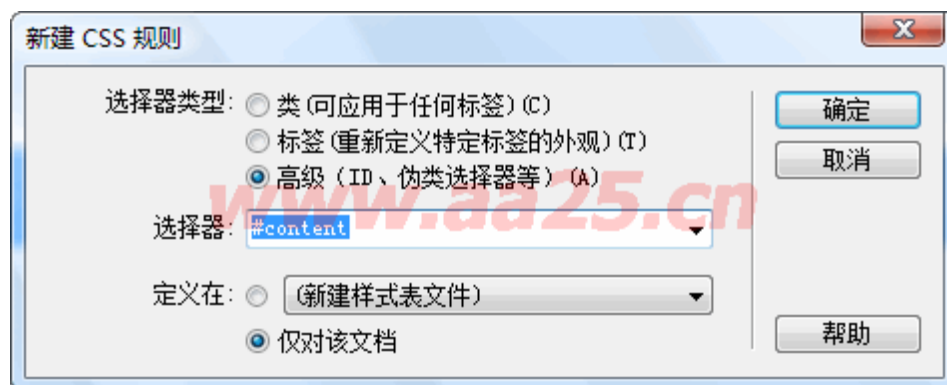
两列固定宽度居中，需要在两列固定宽度的基础上改进，在学一列固定宽度居中时，我们知道让它居中的方法，所以这里需要在这两个 div 之外再加一个父 div：

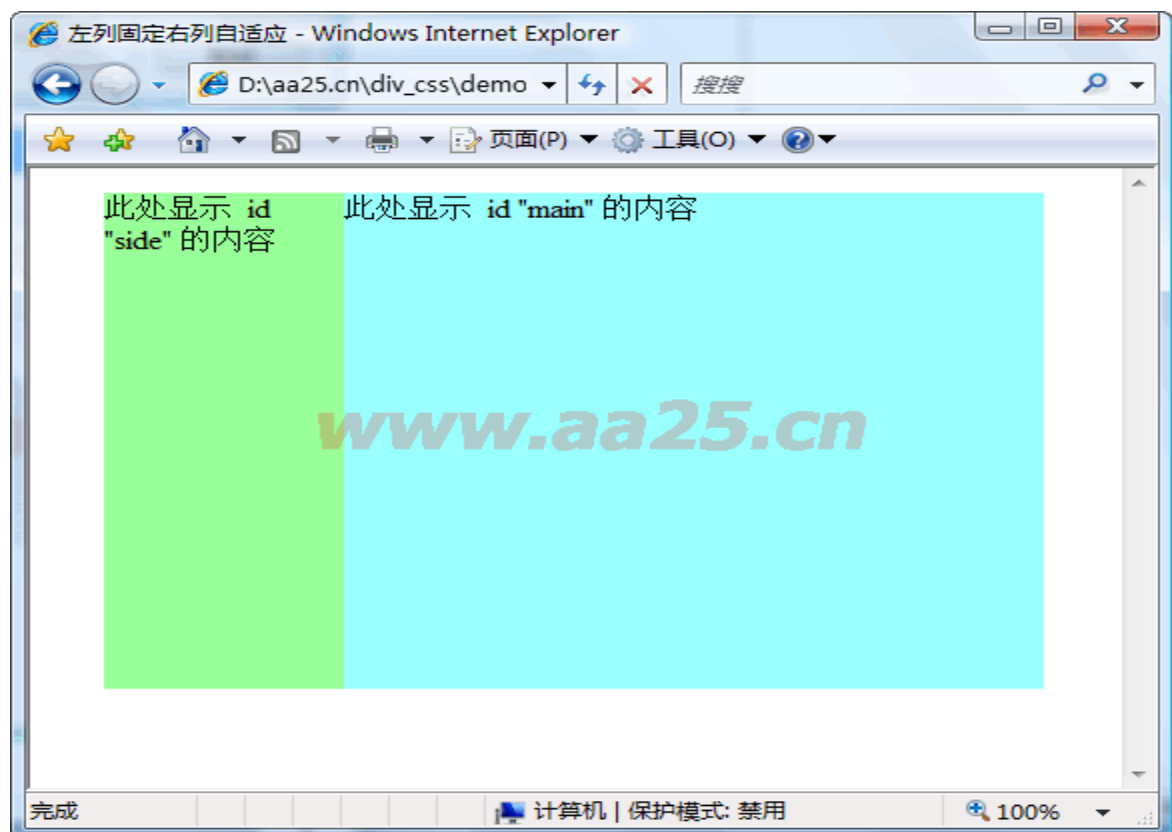
```
<div id="content">  
<div id="side">此处显示 id "side" 的内容</div>  
<div id="main">此处显示 id "main" 的内容</div>  
</div>
```

操作方法：在源代码里选中这两个 div 的代码后，点击工具栏上插入 div 按钮，填写 id 后确定，得到如上的代码



下面就需要设置#content 的样式了，我们知道，#side 的宽度为 120px，#main 宽度为 350px，那么#content 的宽度应该为这两者之和，然后设置#content 居中，那么整体就居中了：





```
<!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>

#content { width:470px; margin:0 auto;}

#side { background: #99FF99; height: 300px; width: 120px; float:
left; }

#main { background: #99FFFF; height: 300px; width: 350px; margin-
left: 120px; }

</style>

</head>

<body>

<div id="content">

    <div id="side">此处显示 id "side" 的内容</div>

    <div id="main">此处显示 id "main" 的内容</div>

</div>

</body>

</html>

```

四、xhtml 的块级元素(div)和内联元素(span)

块级元素：就是一个方块，像段落一样，默认占据一行出现；

内联元素：又叫行内元素，顾名思义，只能放在行内，就像一个单词，不会造成前后换行，起辅助作用。

一般的块级元素诸如段落<p>、标题<h1><h2>...、列表、、表格<table>、表单<form>、DIV<div>和 BODY<body>等元素。而内联元素则如：表单元素<input>、超级链接<a>、图像、 块级元素的显著特点是：每个块级元素都是从一个新行开始显示，而且其后的元素也需另起一行进行显示。

```
8 <body>
9 <h1>这是h1中的内容</h1>
10 <div>这是第一个div内的内容
11 <div>这是第二个div内的内容，<span>这是第二个div中span的内容，</span><em>这是第二个
div中em的内容</em></div>
12 </div>
```

这是h1中的内容

这是第一个div内的内容
这是第二个div内的内容，这是第二个div中span的内容，这是第二个div中em的内容

从上图的例子可以看出，块级元素默认占据一行，相当于在它之前和之后各插入了一个换行；而内联元素 span 没对显示效果造成任何影响，事实也是如此；em 只是让字体变成了斜体，也没有单独占据一行。这就是块级元素和内联元素，正因为有了这些元素，才使我们的网页变得丰富多彩。

如果没有 css 的作用，块元素会顺序以每次另起一行的方式一直往下排。而有了 css 以后，我们可以改变这种 html 的默认布局模式，把块元素摆放到你想要的位置上去。而不是每次都愚蠢的另起一行。也就是说，可以用 css 的 display:inline 将块级元素改变为内联元素，也可以用 display:block 将内联元素改变为块元素。

五、float 属性

回到我们的例子当中，理解了块级元素和内联元素，对理解浮动就容易多了。float 是个重点，希望大家都能理解。上例中用 float 让元素向左浮动，在 CSS 中，任何元素都可以浮动。浮动元素会生成一个块级框，而不论它本身是何种元素；且要指明一个宽度，否则它会尽可能地窄；另外当可供浮动的空间小于浮动元素时，它会跑到下一行，直到拥有足够放下它的空间。

想必大家都用过 word 排过版，当中有个图文混排功能，如下图：

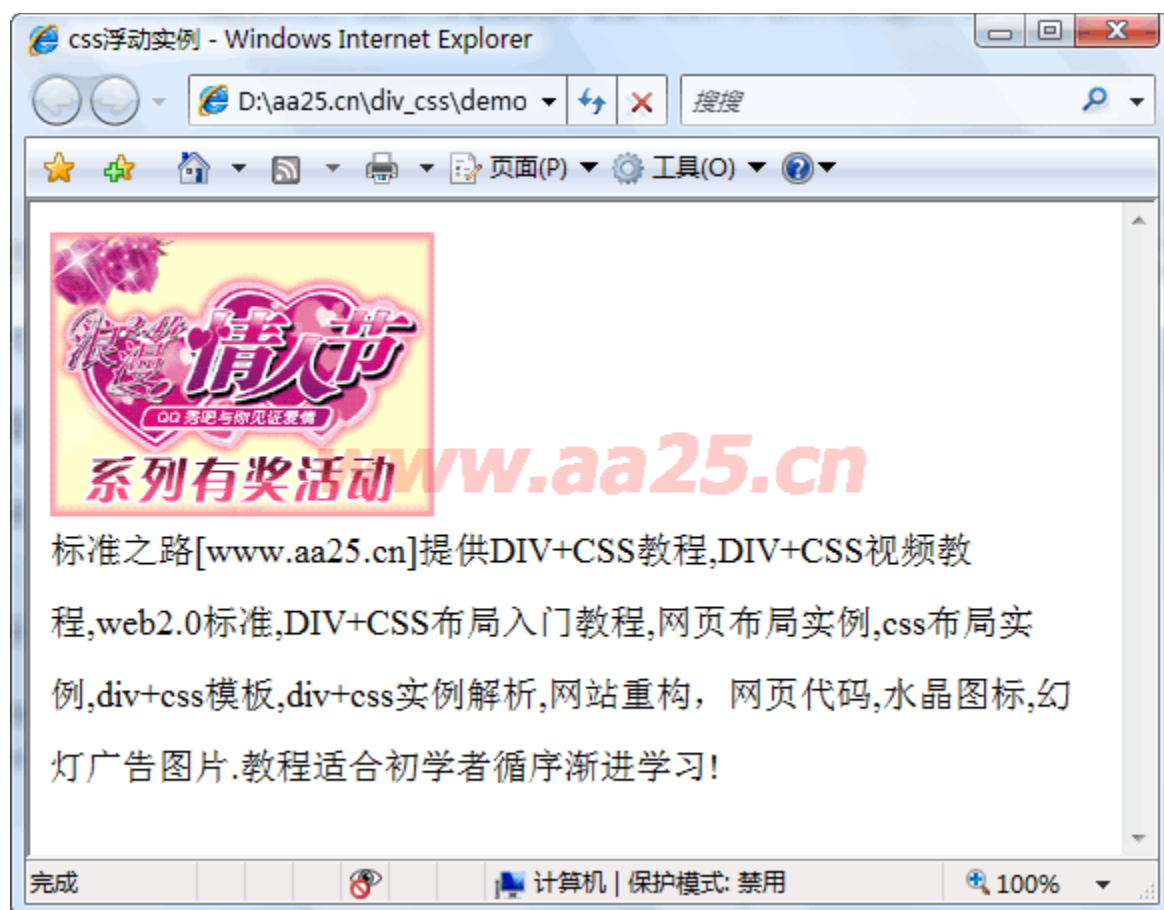


标准之路[www.aa25.cn]提供
DIV+CSS 教程,DIV+CSS 视频教程,web2.0 标准,DIV+CSS 布局入门教程,网页布局实例,css 布局实例,div+css 模板,div+css 实例解析,网站重构,网页代码,水晶图标,幻灯广告图片.教程适合初学者循序渐进学习!↓

这个功能非常类似于 css 中的浮动，下面我们使用 css 来实现这一功能：

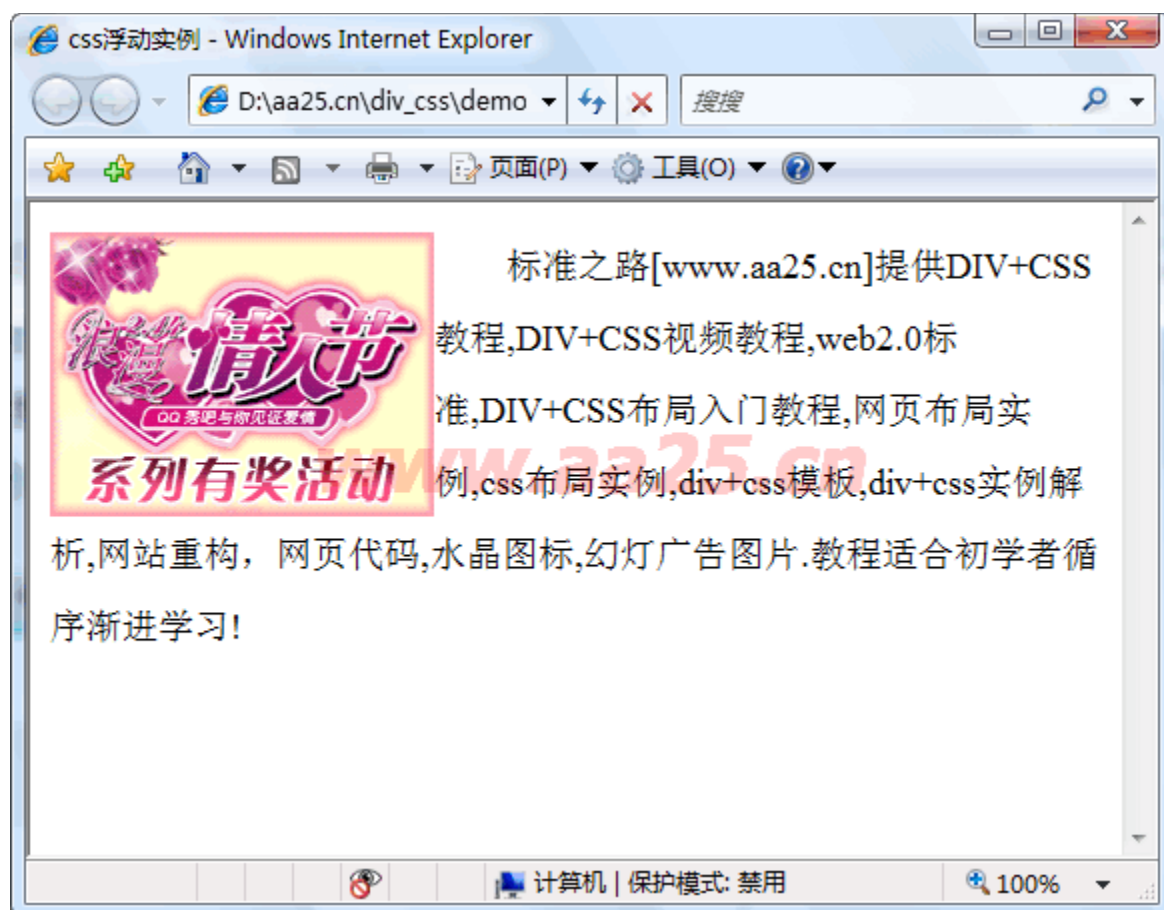
```
<div id="side"></div>  
<div id="main"> 标准之路[www.aa25.cn]提供 DIV+CSS 教程, DIV+CSS 视频教程,  
web2.0 标准, DIV+CSS 布局入门教程, 网页布局实例, css 布局实例, div+css 模板,  
div+css 实例解析, 网站重构, 网页代码, 水晶图标, 幻灯广告图片. 教程适合初  
学者循序渐进学习!</div>
```

创建完以上代码后，在浏览器中预览会显示如下结果：



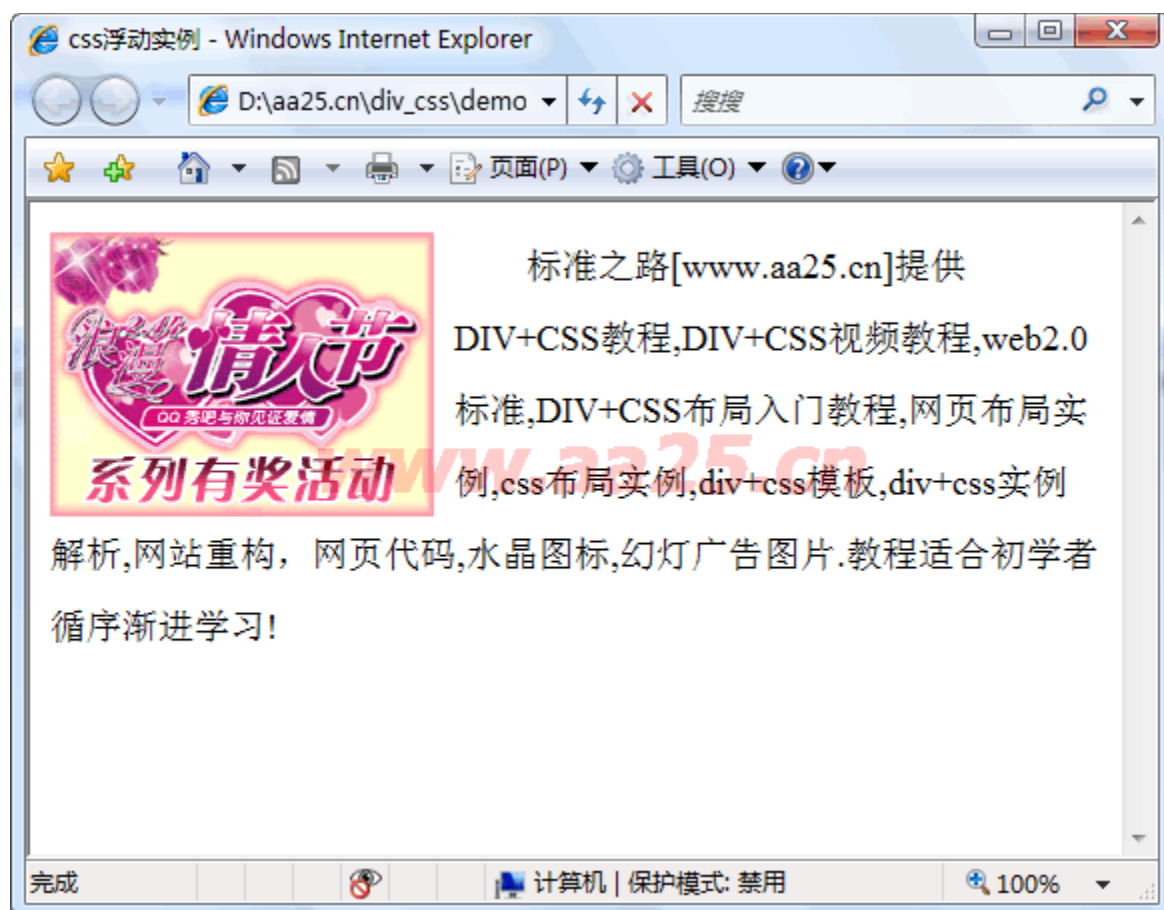
下面用 css 让 side 浮动，再看看效果

```
body { font-size:18px; line-height:200%; }  
#side { float:left;}
```



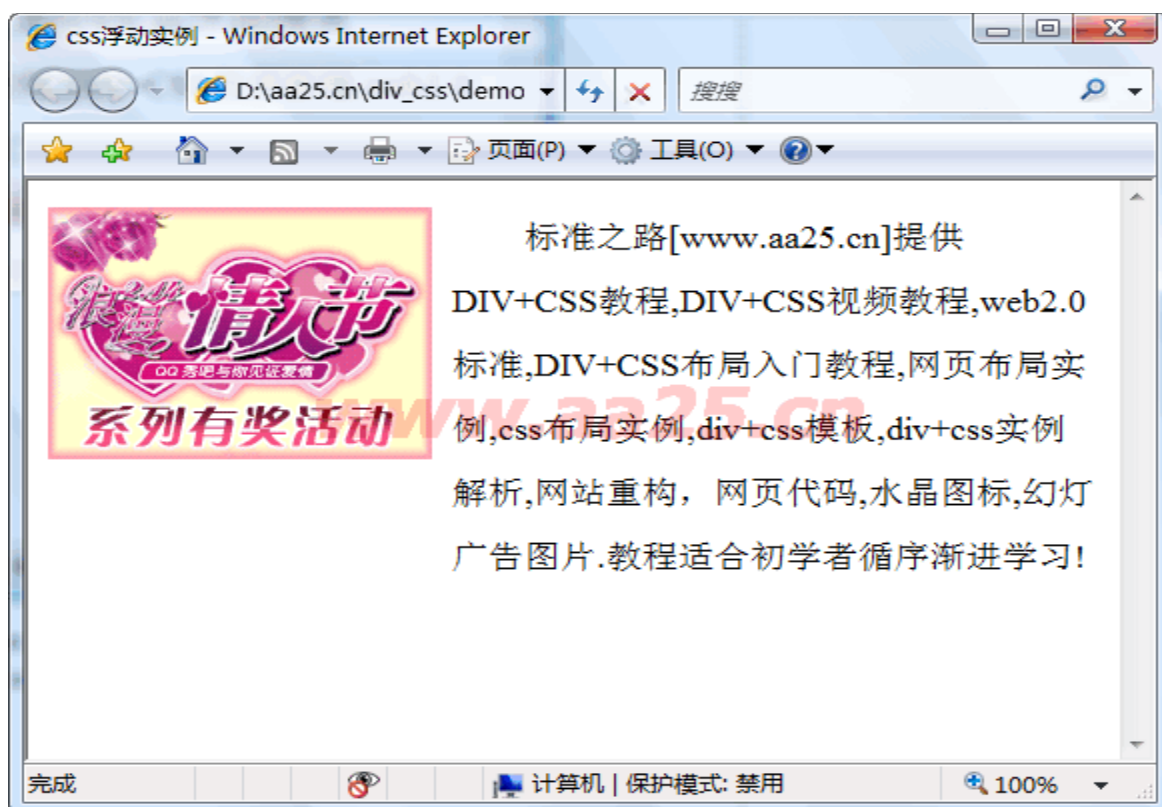
大家看看，是不是和 word 的效果已经很像相了，但这里的文字和图片右侧贴的很近，怎么办呢？刚才已经说了，当元素浮动过之后，需要指定一个宽度，否则它会尽可能窄。那么把 side 的宽度设置为大于图片的宽度，它们中间应该就有空隙了。图片的宽度是 192px，设置 side 为 202px，中间将会有 10px 的空隙了。

```
body { font-size:18px; line-height:200%; }  
#side { float:left; width:202px;}
```



细心的朋友已经发现，上例中#main的div还定义了margin-left:120px;而这里没有定义，但它多出的文字却跑到了图片(#side)的下方，是不是设置margin-left:后，它就不会跑到#side的正文呢？如果你能想到这一点，你的确是太聪明了，事实确实是这样，在css样式表中加上下面一行

```
#main { margin-left:202px;}
```

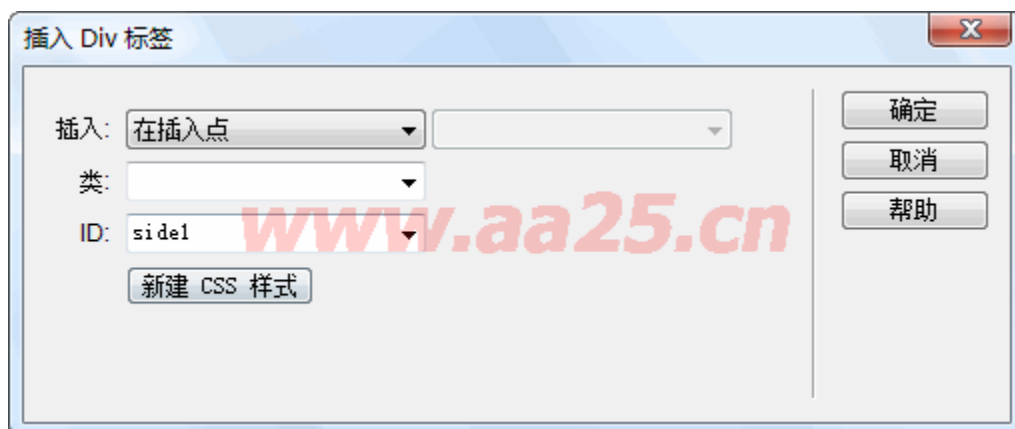


这就是如何应用浮动实现两列布局的原理。那么三列呢？

六、三列自适应宽度

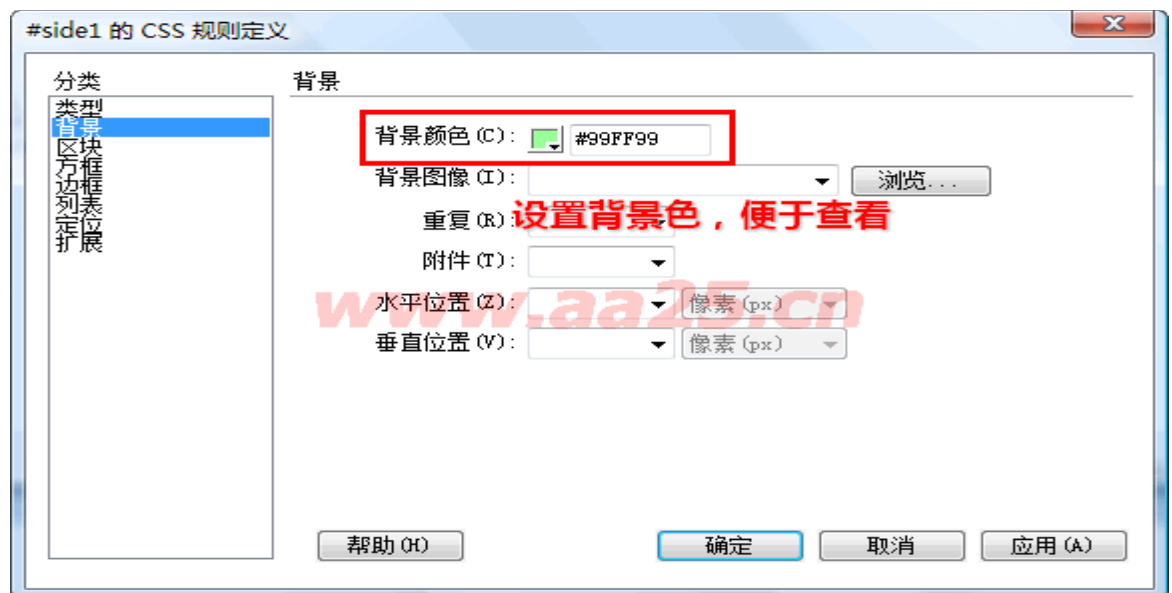
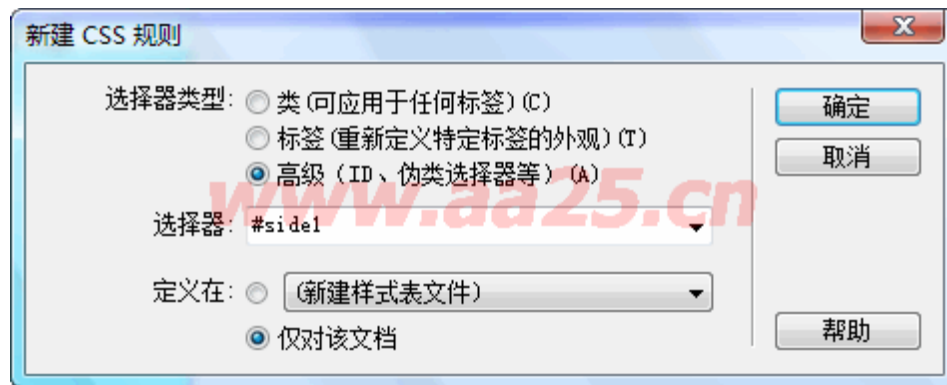
三列自适应宽度，一般常用的结构是左列和右列固定，中间列根据浏览器宽度自适应。下面在二列自适应宽度基础上修改一下

```
<div id="side">此处显示 id "side" 的内容</div>
<div id="side1">此处显示 id "side1" 的内容</div>
<div id="main">此处显示 id "main" 的内容</div>
```



增加以下 css 样式：

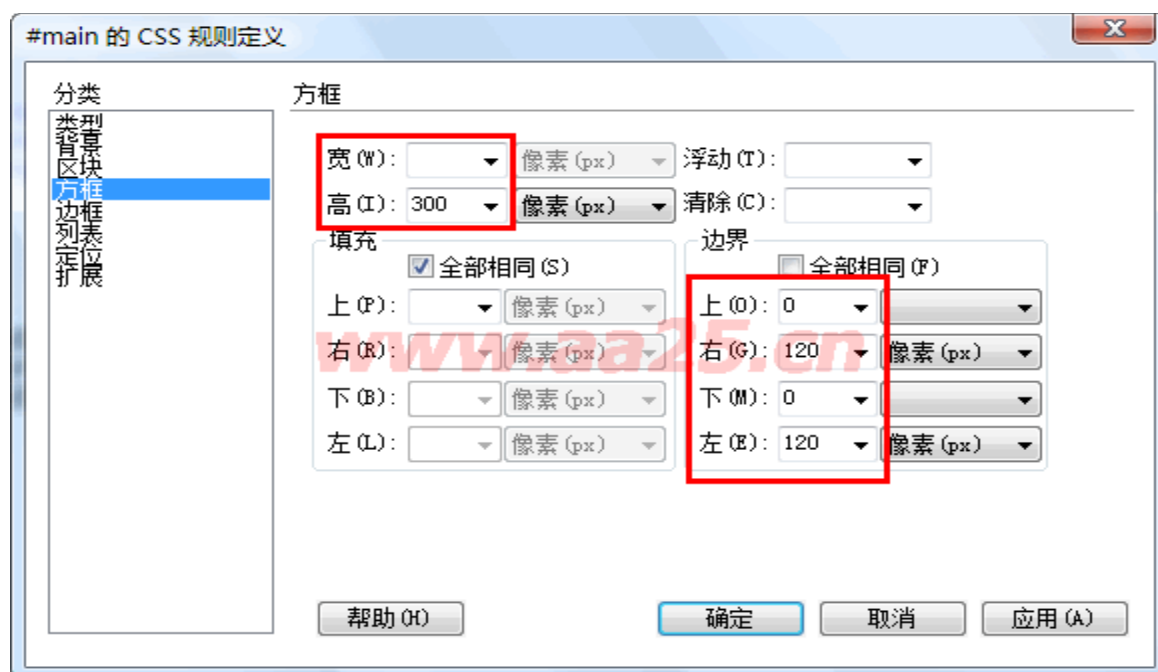
```
#side1 { background: #99FF99; height: 300px; width: 120px; float: right; }
```



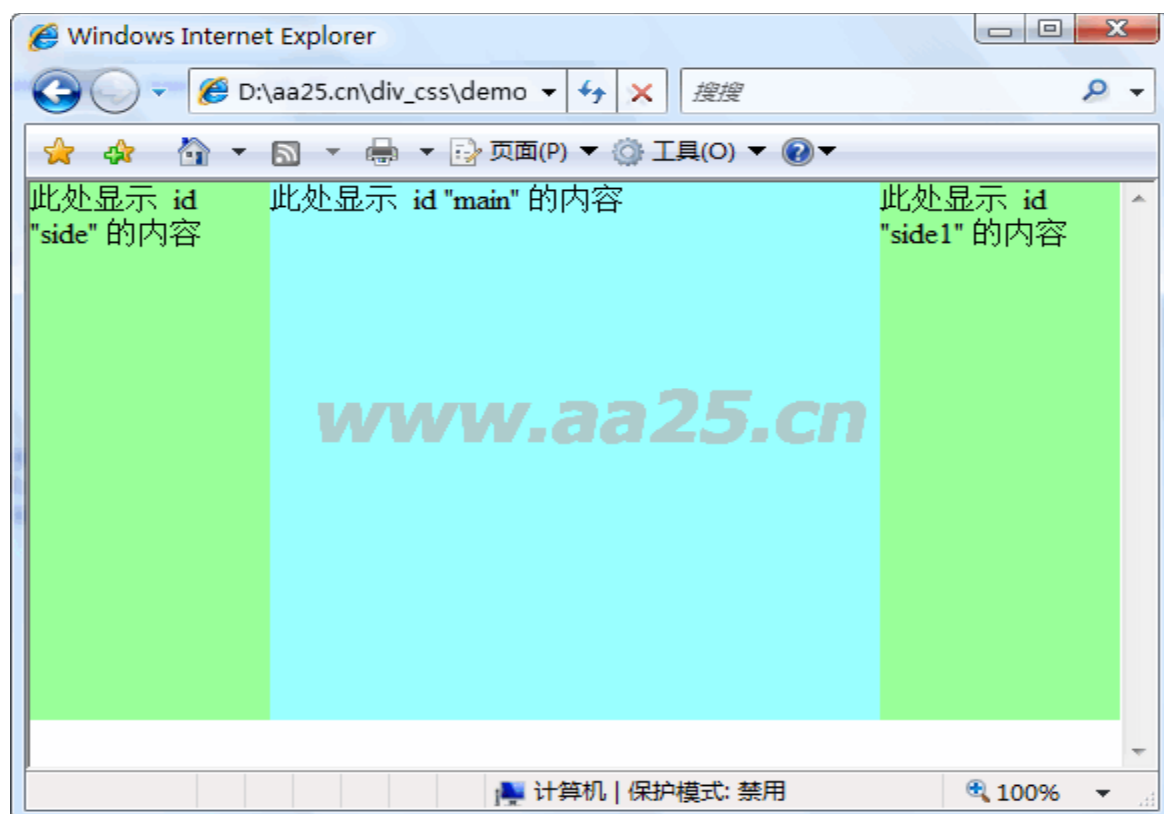


把原来#main 样式的 width:70%去掉，然后设置左右外边距各 120px，让出左右列的宽度

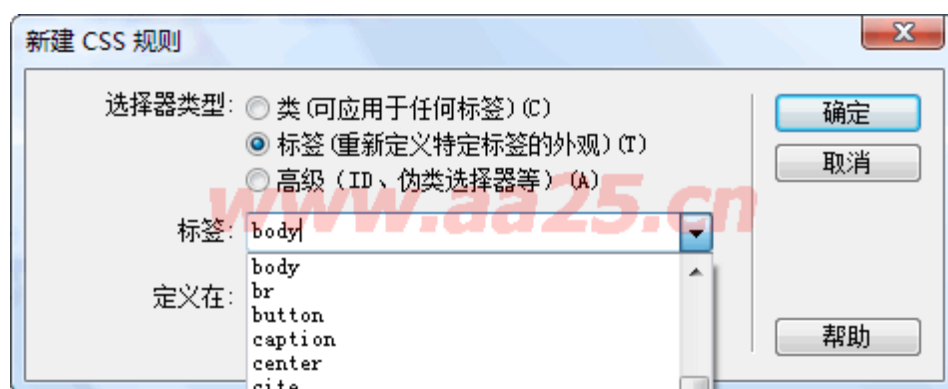
```
#main { background: #99FFFF; height: 300px; margin:0 120px; }
```

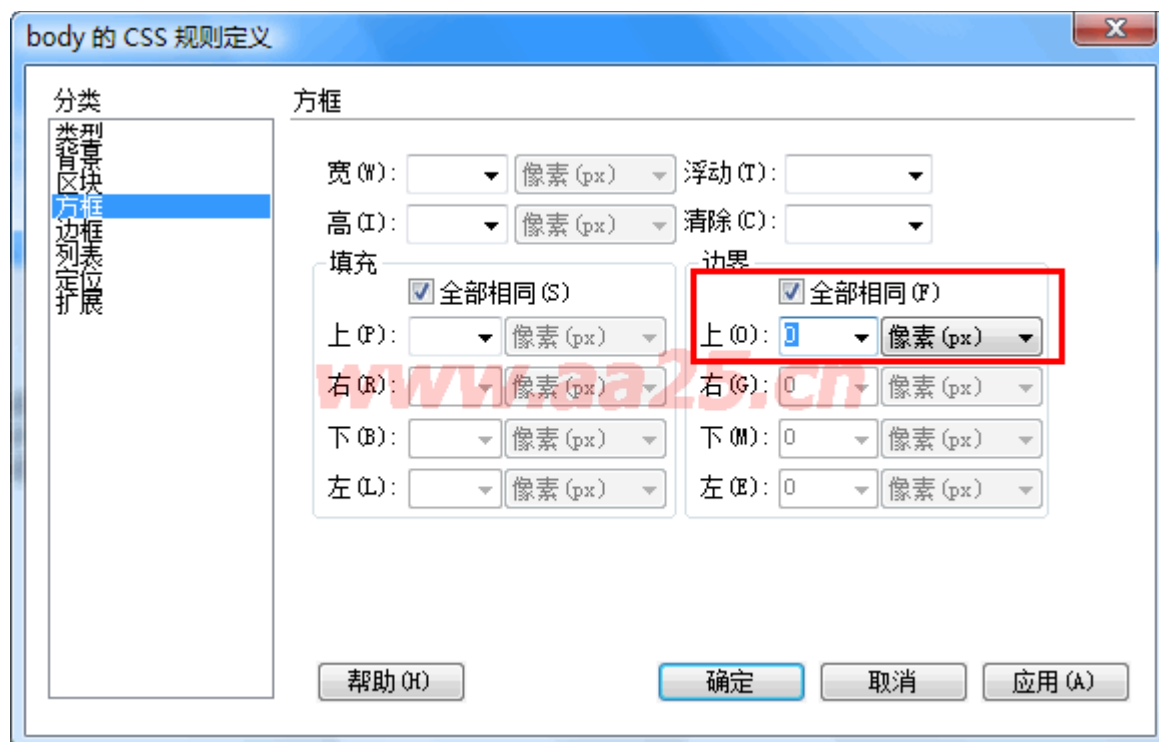


预览一下效果如下：



也许预览出来的效果和我的不一样，别急，还有一项设置，默认 body 是有外边距的，所以这里还得设置 body 的外边距为 0；点击 css 面板上的新建按钮，然后在新建面板中选择：标签（重定义标签外观），然后选择 body，设置 body 的边界为 0 即可。





```
<!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>

body { margin:0;}

#side { background: #99FF99; height: 300px; width: 120px; float:
left; }

#side1 { background: #99FF99; height: 300px; width: 120px; float:
right; }

#main { background: #99FFFF; height: 300px; margin:0 120px; }

</style>

</head>

<body>

<div id="side">此处显示 id "side" 的内容</div>

<div id="side1">此处显示 id "side1" 的内容</div>
```

```
<div id="main">此处显示 id "main" 的内容</div>
```

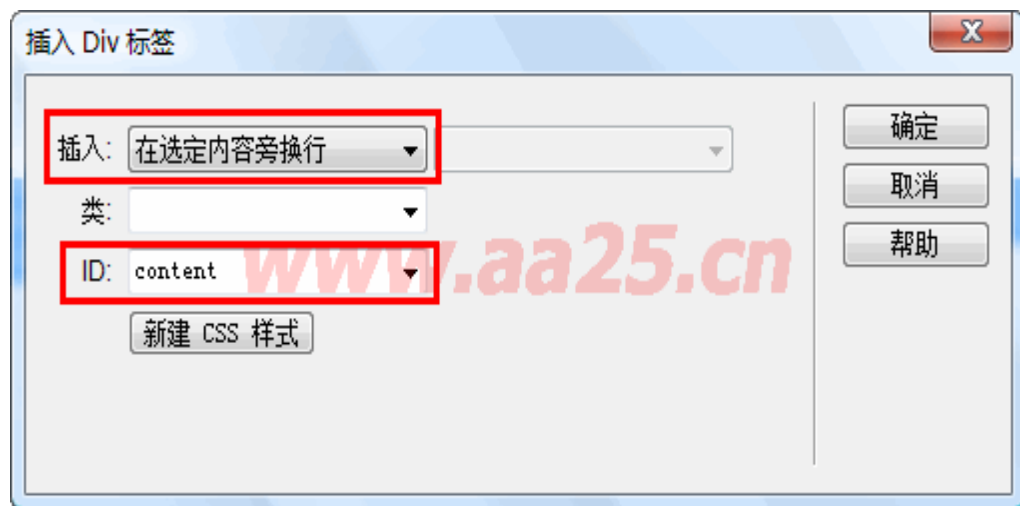
```
</body>
```

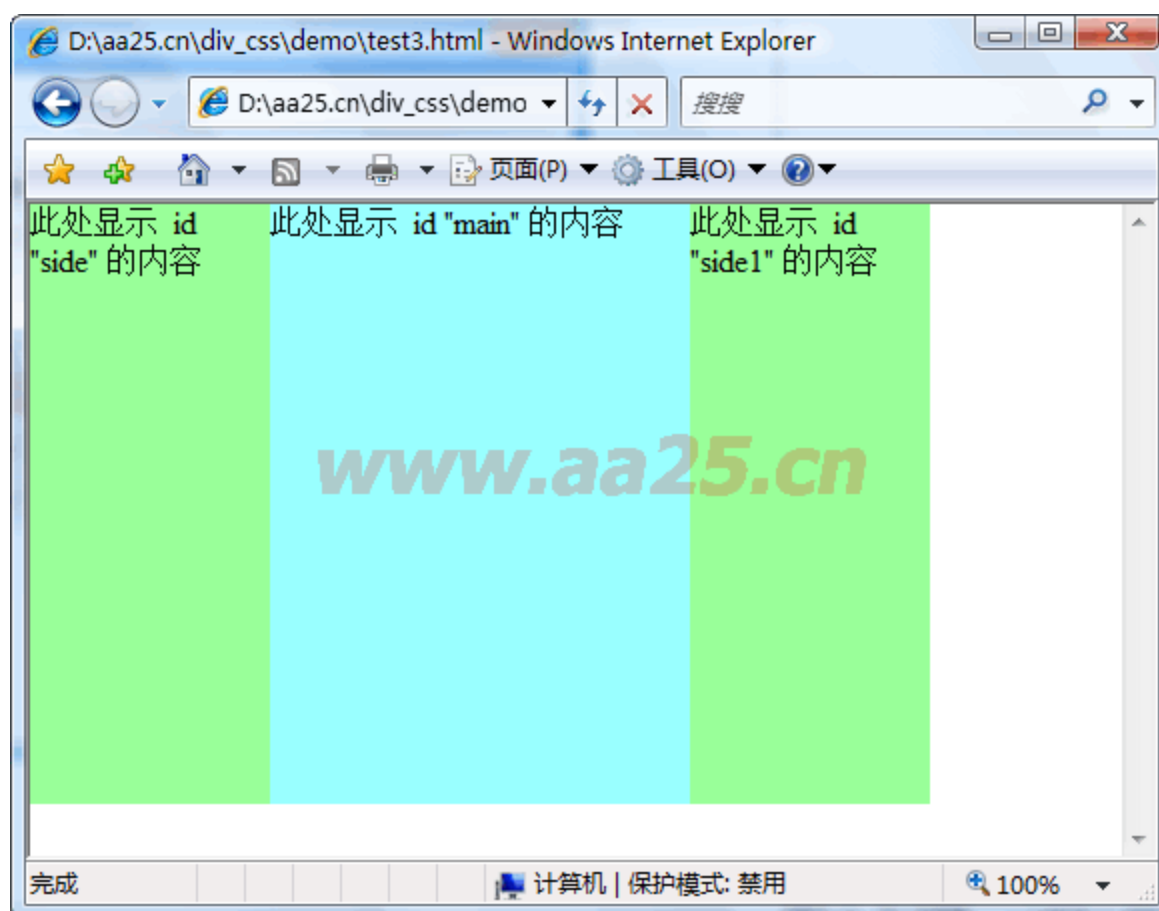
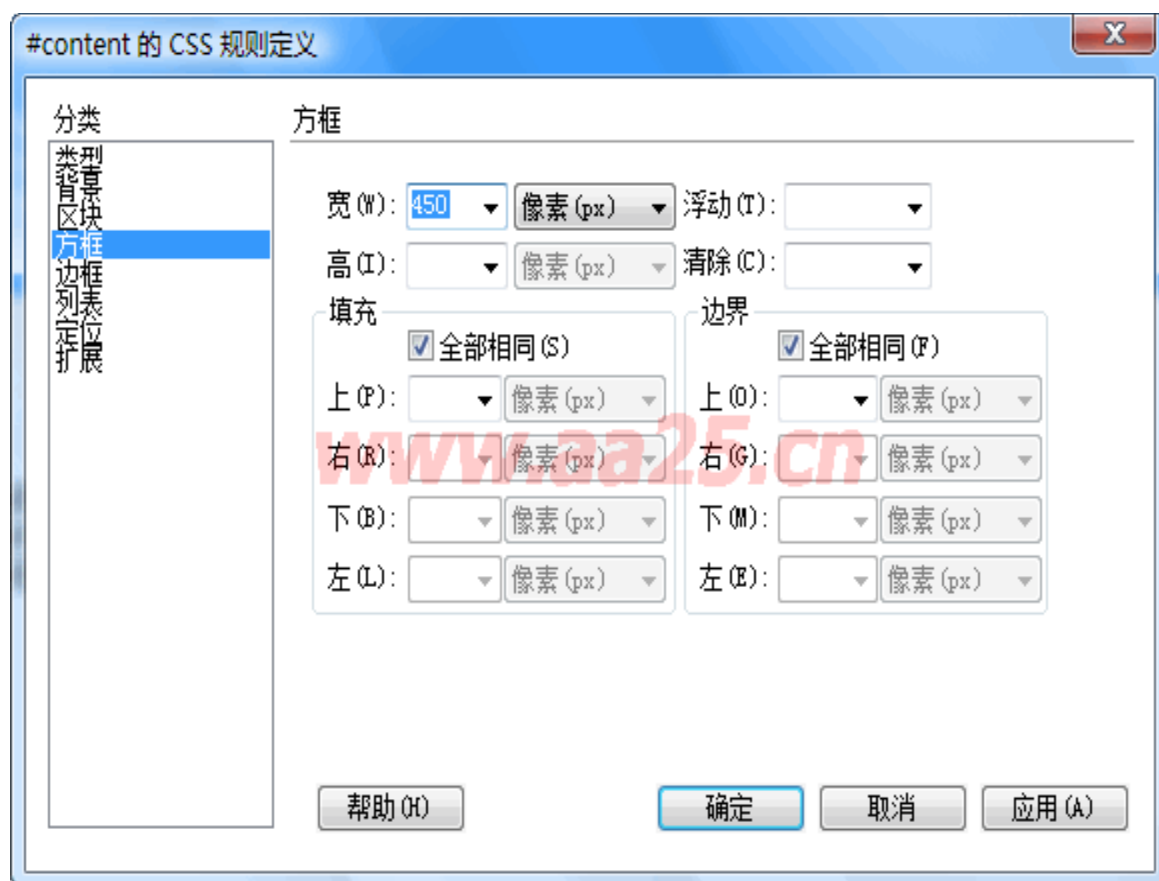
```
</html>
```

七、三列固定宽度

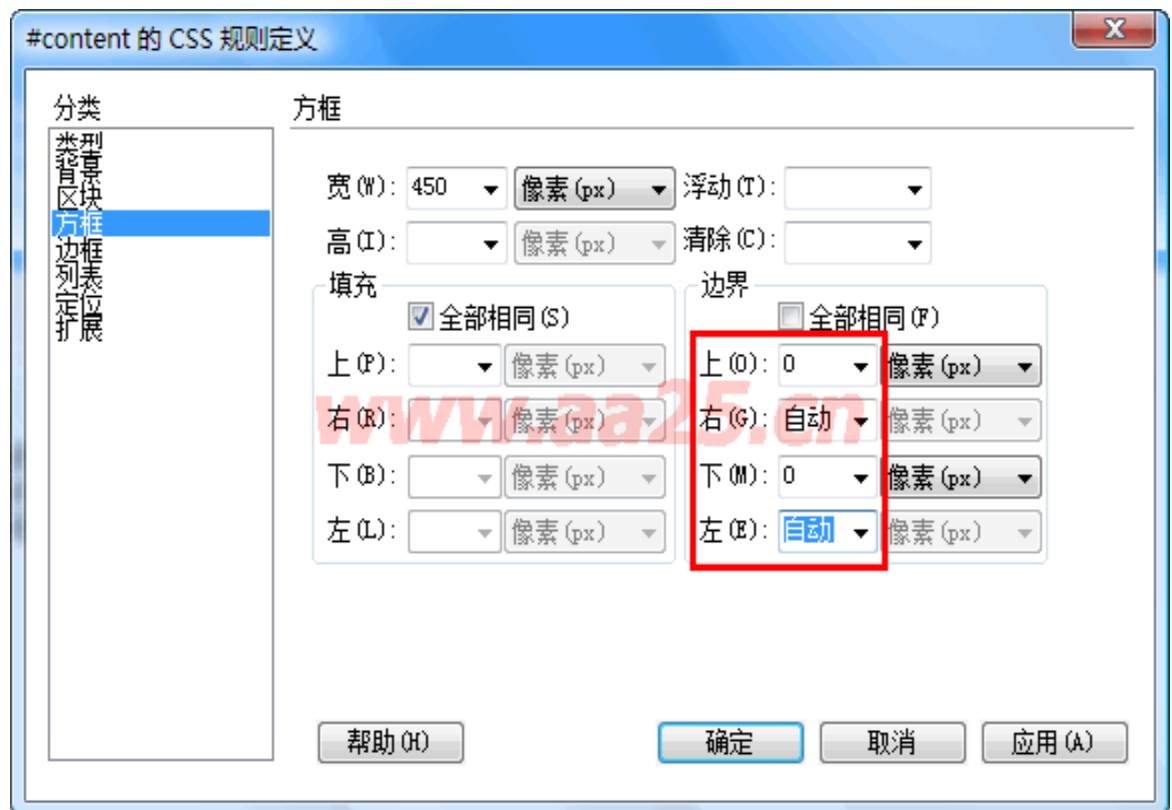
三列固定宽度可以在三列自适应基础上添加一个父 div，并设置这个 div 的宽度即可，如下，添加一个 id 为 content 的父容器。

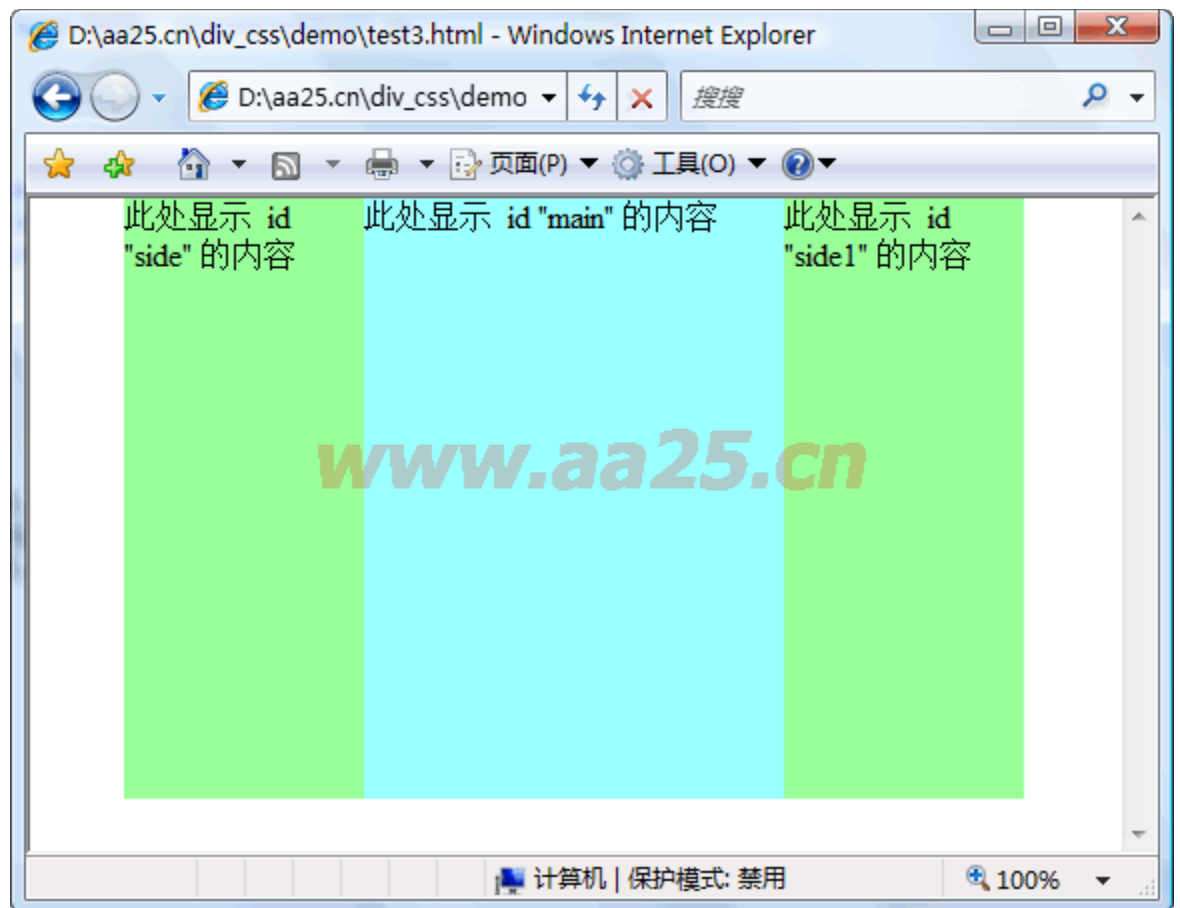
在源代码里选中这三个 div，然后点击工具栏上的“插入 div 标签”按钮，这时弹出的窗口插入项会默认为：在选定的内容旁换行，输入 id 为 content，然后给这个 div 定义个宽度





三列固定宽度出来了，要想实现三列固定宽度居中就更方便了，只需设置
#content {margin:0 auto;}即可





```
<!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>

body { margin:0;}

#content { width:470px; margin:0 auto;}

#side { background: #99FF99; height: 300px; width: 120px; float:
left; }

#side1 { background: #99FF99; height: 300px; width: 120px; float:
right; }

#main { background: #99FFFF; height: 300px; margin:0 120px; }

</style>
```

```
</head>
<body>
<div id="content">
  <div id="side">此处显示 id "side" 的内容</div>
  <div id="side1">此处显示 id "side1" 的内容</div>
  <div id="main">此处显示 id "main" 的内容</div>
</div>
</body>
</html>
```

八、IE6 的 3 像素 bug

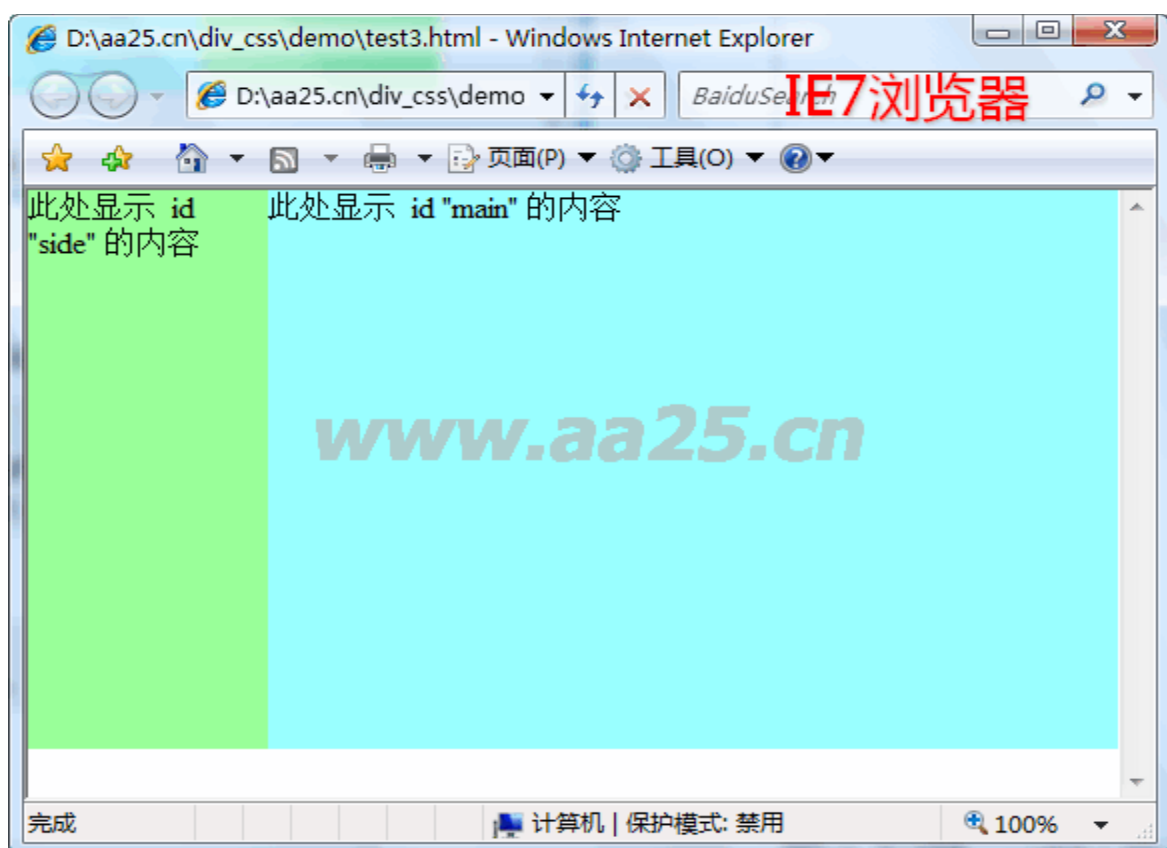
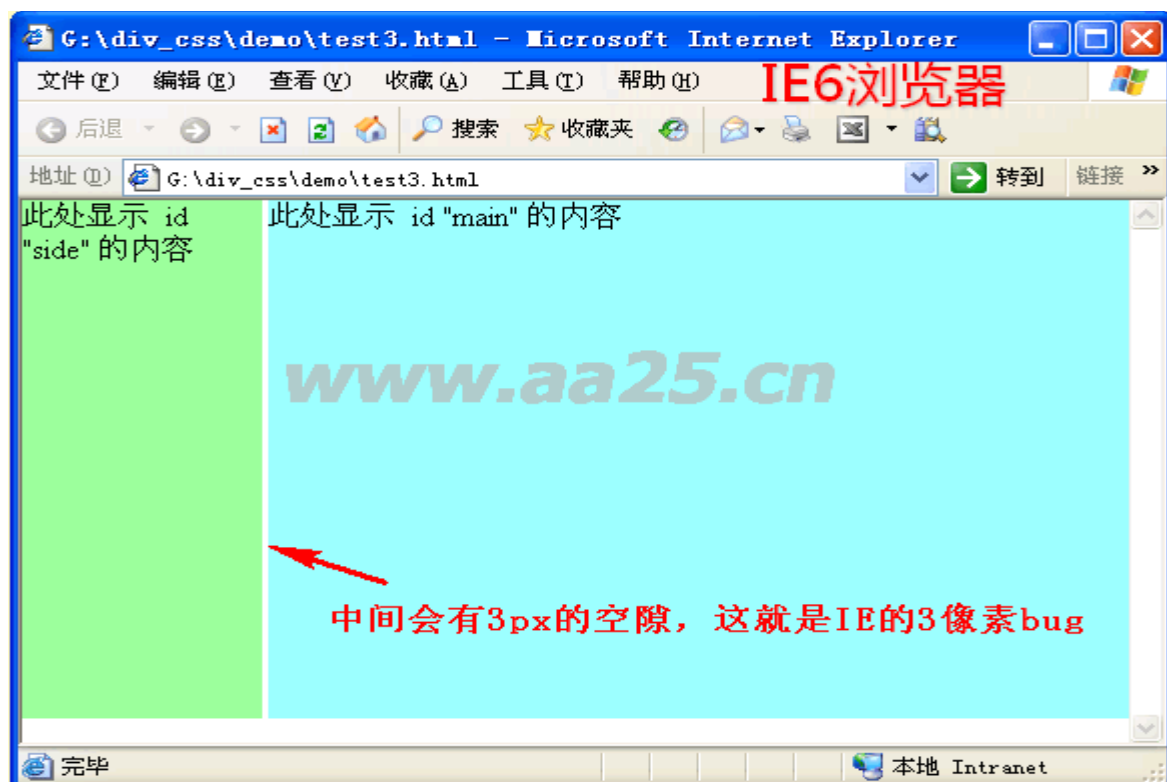
3 像素 bug 是 IE6 的一个著名的 bug，当浮动元素与非浮动元素相邻时，这个 3 像素的 Bug 就会出现。看下面这个左列固定，右列液态的例子，css 代码如下：

```
body { margin:0;}
#side { float: left; background:#99FF99; height: 300px; width:
120px;}
#main { background: #99FFFF; height: 300px;}
```

html 代码如下：

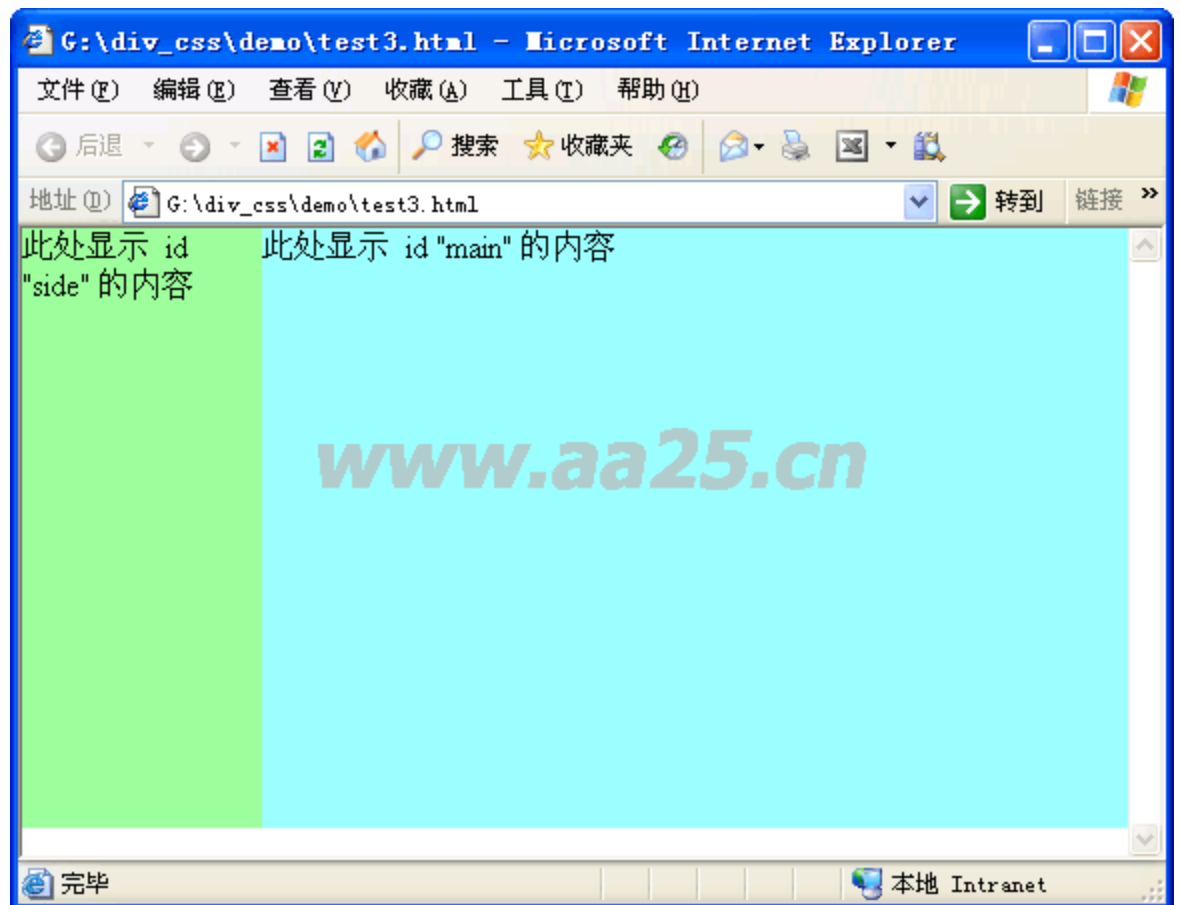
```
<div id="side">此处显示 id "side" 的内容</div>
<div id="main">此处显示 id "main" 的内容</div>
```

下面看看 IE6 和 IE7 中的显示效果：



从截图中明显看出，IE6 会在两个 div 中间加上 3px 的空隙，那么要解决这个问题，请在#side 上加上 `_margin-right:-3px;`；记住，前边加上一下划线，这样这个样式专门针对 IE6 生效。IE7 和 FF 下还会正常显示。

```
body { margin:0;}
#side { float: left; background:#99FF99; height: 300px; width: 120px;
       _margin-right:-3px;}
#main { background: #99FFFF; height: 300px; }
```



看看，是不是问题已经解决了。但它不能通过 W3C 验证。当两列固定宽度时，最好把#main 也固定宽度且向右浮动，这样就可以避免 IE6 的 3 像素 bug 了。