# 浅谈CSS布局

CSS包含3种基本的布局模型，用英文概括为：Flow、Layer 和 Float。

在网页中，元素有三种布局模型：

1、流动模型（Flow）

2、浮动模型 (Float)

3、层模型（Layer）

## 流动模型

流动（Flow）是默认的网页布局模式。也就是说网页在默认状态下的 HTML 网页元素都是根据流动模型来分布网页内容的。

流动布局模型具有2个比较典型的特征：

第一点，块状元素都会在所处的包含元素内自上而下按顺序垂直延伸分布，因为在默认状态下，块状元素的宽度都为100%。实际上，块状元素都会以行的形式占据位置。

第二点，在流动模型下，内联元素都会在所处的包含元素内从左到右水平分布显示。（内联元素可不像块状元素这么霸道独占一行）。

## 浮动模型

块状元素这么霸道都是独占一行，如果现在我们想让两个块状元素并排显示，怎么办呢？不要着急，设置元素浮动就可以实现这一愿望。

任何元素在默认情况下是不能浮动的，但可以用 CSS 定义为浮动，如 div、p、table、img 等元素都可以被定义为浮动。如下代码可以实现两个 div 元素一行显示。

div{

width:200px;

height:200px;

border:2px red solid;

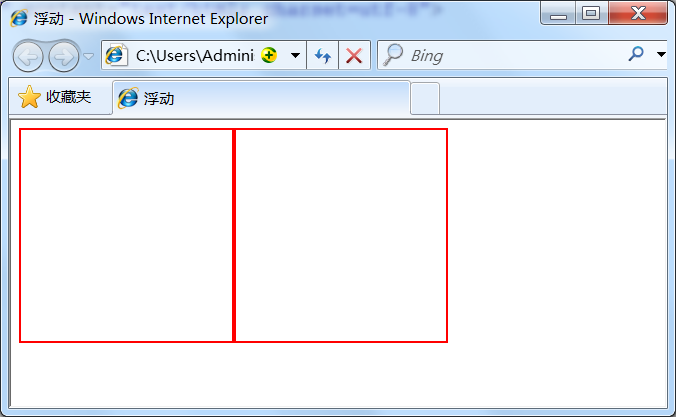
float:left;

}

<div id="div1"></div>

<div id="div2"></div>

效果图



当然你也可以同时设置两个元素右浮动也可以实现一行显示。

div{

width:200px;

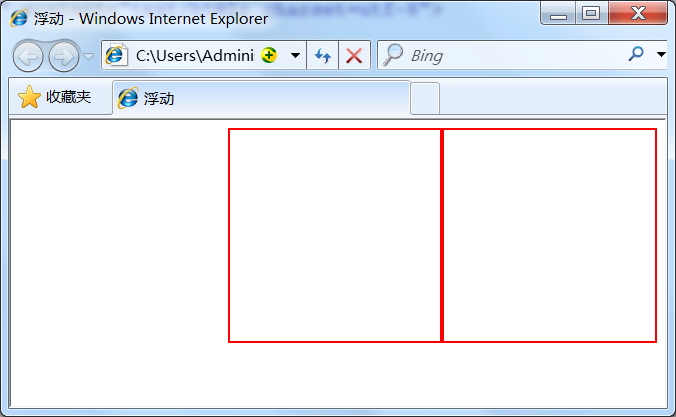
height:200px;

border:2px red solid;

float:right;

}

效果图



又有小伙伴问了，设置两个元素一左一右可以实现一行显示吗？当然可以：

div{

width:200px;

height:200px;

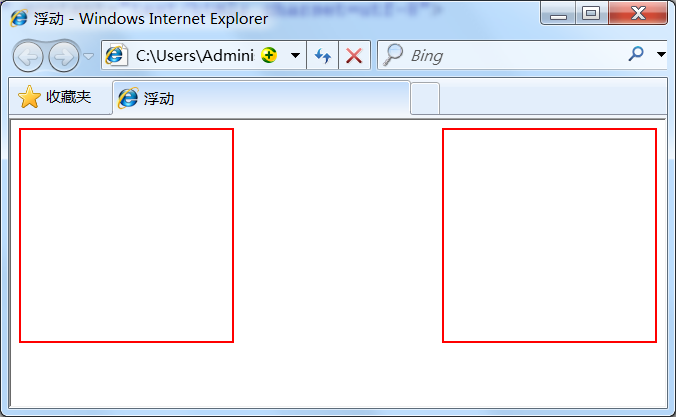
border:2px red solid;

}

#div1{float:left;}

#div2{float:right;}

效果图



## 层模型

什么是层布局模型？层布局模型就像是图像软件PhotoShop中非常流行的图层编辑功能一样，每个图层能够精确定位操作，但在网页设计领域，由于网页大小的活动性，层布局没能受到热捧。但是在网页上局部使用层布局还是有其方便之处的。下面我们来学习一下html中的层布局。

如何让html元素在网页中精确定位，就像图像软件PhotoShop中的图层一样可以对每个图层能够精确定位操作。CSS定义了一组定位（positioning）属性来支持层布局模型。

层模型有三种形式：

1、**绝对定位**(position: absolute)

2、**相对定位**(position: relative)

3、**固定定位**(position: fixed)

### 绝对定位

如果想为元素设置层模型中的绝对定位，需要设置position:absolute(表示绝对定位)，这条语句的作用将元素从文档流中拖出来，然后使用left、right、top、bottom属性相对于其最接近的一个具有定位属性的父包含块进行绝对定位。如果不存在这样的包含块，则相对于body元素，即相对于浏览器窗口。

如下面代码可以实现div元素相对于浏览器窗口向右移动100px，向下移动50px。

div{

width:200px;

height:200px;

border:2px red solid;

position:absolute;

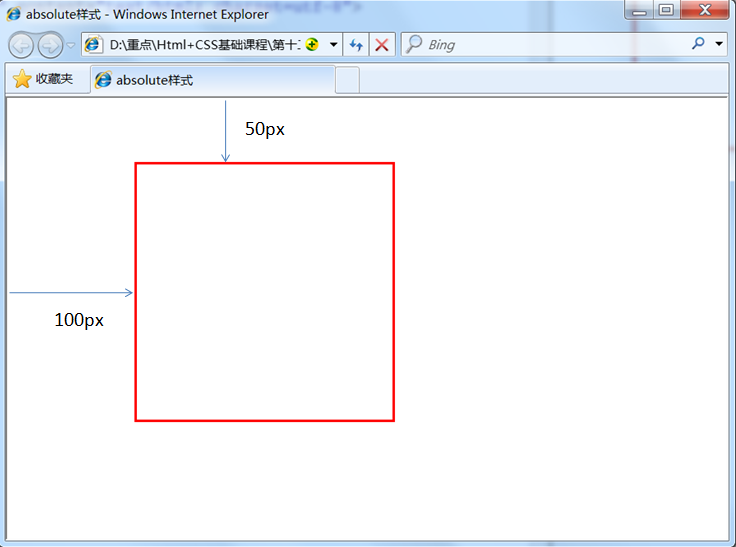
left:100px;

top:50px;

}

<div id="div1"></div>

效果如下：



### 相对定位

如果想为元素设置层模型中的相对定位，需要设置position:relative（表示相对定位），它通过left、right、top、bottom属性确定元素在正常文档流中的偏移位置。相对定位完成的过程是首先按static(float)方式生成一个元素(并且元素像层一样浮动了起来)，然后相对于以前的位置移动，移动的方向和幅度由left、right、top、bottom属性确定，偏移前的位置保留不动。

如下代码实现相对于以前位置向下移动50px，向右移动100px;

#div1{

width:200px;

height:200px;

border:2px red solid;

position:relative;

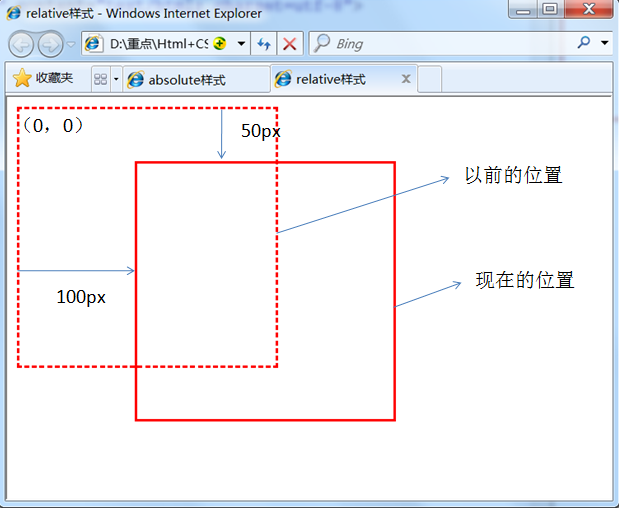
left:100px;

top:50px;

}

<div id="div1"></div>

效果图：



什么叫做“偏移前的位置保留不动”呢？

代码如下：

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>relative样式</title>

<style type="text/css">

#div1{

width:200px;

height:200px;

border:2px red solid;

position:relative;

left:100px;

top:50px;

}

</style>

</head>

<body>

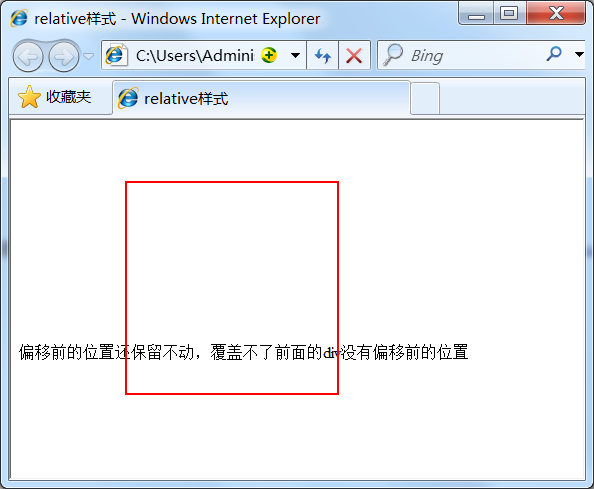
<div id="div1"></div>

<span>偏移前的位置还保留不动，覆盖不了前面的div没有偏移前的位置</span>

</body>

</html>

效果图：



从效果图中可以明显的看出，虽然div元素相对于以前的位置产生了偏移，但是div元素以前的位置还是保留着，所以后面的span元素是显示在了div元素以前位置的后面。

### 固定定位

fixed：表示固定定位，与absolute定位类型类似，但它的相对移动的坐标是视图（屏幕内的网页窗口）本身。由于视图本身是固定的，它不会随浏览器窗口的滚动条滚动而变化，除非你在屏幕中移动浏览器窗口的屏幕位置，或改变浏览器窗口的显示大小，因此固定定位的元素会始终位于浏览器窗口内视图的某个位置，不会受文档流动影响，这与background-attachment:fixed;属性功能相同。以下代码可以实现相对于浏览器视图向右移动100px，向下移动50px。并且拖动滚动条时位置固定不变。

#div1{

width:200px;

height:200px;

border:2px red solid;

position:fixed;

left:100px;

top:50px;

}

<div id="div1"></div>

### Relative与Absolute组合使用

前面讲到绝对定位的方法：使用position:absolute可以实现被设置元素相对于浏览器（body）设置定位以后，大家有没有想过可不可以相对于其它元素进行定位呢？答案是肯定的，当然可以。使用position:relative来帮忙，但是必须遵守下面规范：

1、参照定位的元素必须是相对定位元素的前辈元素。

<div id="box1"><!--参照定位的元素-->

<div id="box2">相对参照元素进行定位</div><!--相对定位元素-->

</div>

2、参照定位的元素必须加入position:relative。

#box1{

width:200px;

height:200px;

position:relative;

}

3、定位元素加入position:absolute，便可以使用top、bottom、left、right来进行偏移定位了。

#box2{

position:absolute;

top:20px;

left:30px;

}