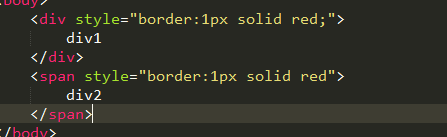
# 块级元素内联元素

块级元素如div默认是占用100%的宽度  
而内联元素如span的默认宽度由其内容的宽度决定

实验代码



实验结果



# Css盒子模型（也叫框模型）

在讲到边距的时候不得不提css盒子模型了,所有html元素都可以当作盒子，真正决定一个元素的表现形式，是由其盒子模型决定的，盒子模型包括了：内容(content)也叫元素（element）、填充(padding)、边框(border)、边界(margin)



其中我们主要要来了解的是margin和padding,和content的大小。

margin为元素外边距，表示的是元素边框和元素边框之间的距离。

Padding为元素与边框之间的距离

## Margin：元素外边距

元素外边距包括：

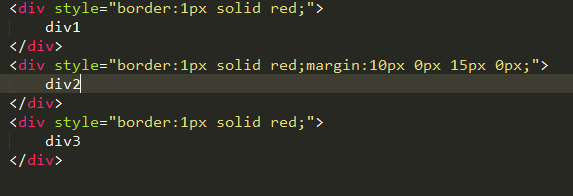
margin-left: 左外边距   
margin-right: 右外边距   
margin-top: 上外边距   
margin-bottom: 下外边距

可以简写外边距属性

如margin:上 右 左 下

是按照上右下左依此设置的

实验代码



实验效果

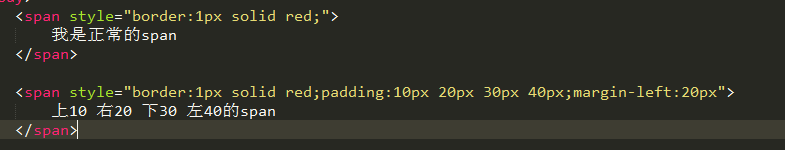


## padding：元素内边距

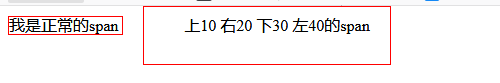
同样元素内边距包括：

padding-left: 左内边距   
padding-right: 右内边距   
padding-top: 上内边距   
padding-bottom: 下内边距   
padding: 上 右 下 左

实验代码



实验效果

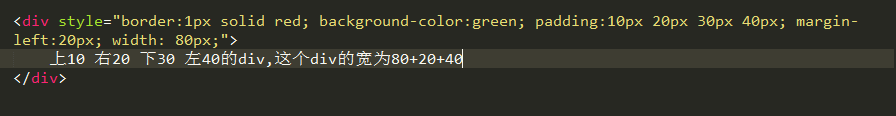


## 内容（元素）大小设置（内联元素会忽略内容大小的设置）

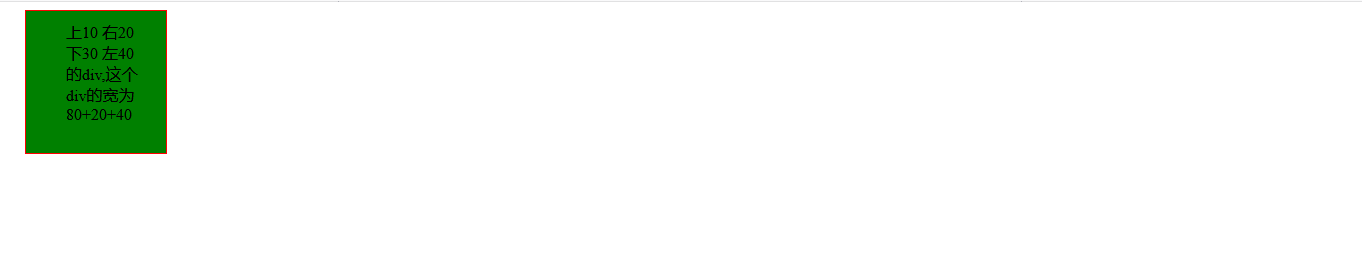
宽度：width

高度：height

实验代码



实验效果



# 定位属性

position属性指定一个元素（静态的，相对的，绝对或固定）的定位方法的类型。

常见的几种属性值：

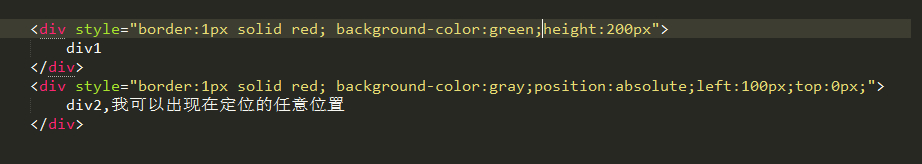
[absolute](http://www.runoob.com/css/css-positioning.html#position-absolute)：生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定，定位使元素的位置与文档流无关，因此不占据空间，与其他元素重叠。

[fixed](http://www.runoob.com/css/css-positioning.html#position-fixed)：生成固定定位的元素，相对于浏览器窗口进行定位。元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。定位使元素的位置与文档流无关，因此不占据空间，与其他元素重叠。

[relative](http://www.runoob.com/css/css-positioning.html#position-relative)：生成相对定位的元素，相对于其正常位置进行定位。因此，"left:20" 会向元素的 LEFT 位置添加 20 像素。但是其原本所占的空间不会改变，所以经常用来作为绝对定位元素的容器块。

[static](http://www.runoob.com/css/css-positioning.html#position-static)：默认值。没有定位，元素出现在正常的流中（忽略 top, bottom, left, right 或者 z-index 声明，不受这些属性的影响）。

实例代码



实验效果

