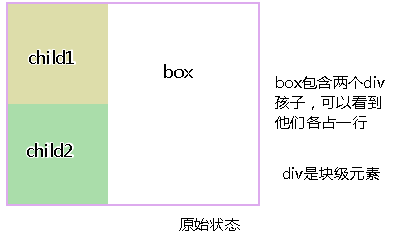
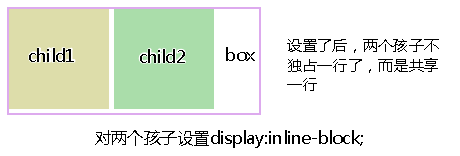
**1.解释一下display的几个常用的属性值，inline ， block， inline-block**

* inline（行内元素）:
  1. 使元素变成行内元素，拥有行内元素的特性，即可以与其他行内元素共享一行，不会独占一行.
  2. 不能更改元素的height，width的值，大小由内容撑开.
  3. 可以使用padding上下左右都有效，margin只有left和right产生边距效果，但是top和bottom就不行.
* block（块级元素）:
  1. 使元素变成块级元素，独占一行，在不设置自己的宽度的情况下，块级元素会默认填满父级元素的宽度.
  2. *能够改变元素的height，width的值.*
  3. *可以设置padding，margin的各个属性值，top，left，bottom，right都能够产生边距效果.*
* inline-block（融合行内于块级）:
  1. 结合了inline与block的一些特点，结合了上述inline的第1个特点和block的第2,3个特点.
  2. 用通俗的话讲，就是不独占一行的块级元素。如图:

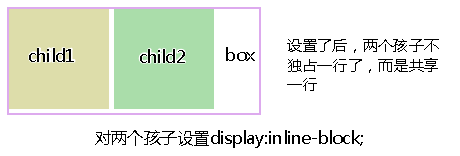
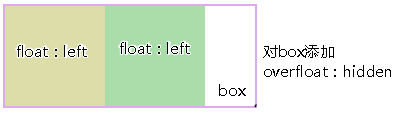
图一:  
  
图二：  


两个图可以看出，display：inline-block后块级元素能够在同一行显示，有人这说不就像浮动一样吗。没错，display：inline-block的效果几乎和浮动一样，但也有不同，接下来讲一下inline-block和浮动的比较。

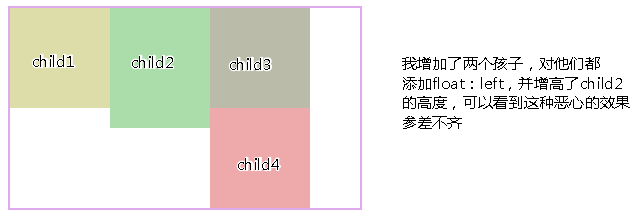
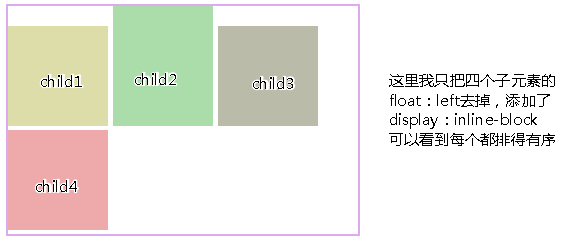
**2.inline-block布局 vs 浮动布局**

**a.不同之处：**对元素设置display：inline-block ，元素不会脱离文本流，而float就会使得元素脱离文本流，且还有父元素高度坍塌的效果

**b.相同之处：**能在某程度上达到一样的效果

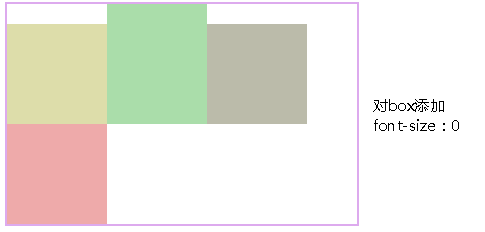
　　我们先来看看这两种布局：  
图一：display：inline-block  
  
图二：对两个孩子使用float：left，我在上一篇浮动布局讲过，这是父元素会高度坍塌，所以要闭合浮动，对box使用overflow：hidden，效果如下：  


>>乍一看两个都能做到几乎相同的效果，（仔细看看display：inline-block中有间隙问题，这个留到下面再讲）

**c.浮动布局不太好的地方：**参差不齐的现象，我们看一个效果：  
图三：   
  
图四：   
  
>>从图3,4可以看出浮动的局限性在于，若要元素排满一行，换行后还要整齐排列，就要子元素的高度一致才行，不然就会出现图三的效果，而inline-block就不会。

**3.inline-block存在的小问题：**

**a**.上面可以看到用了display:inline-block后，存在间隙问题，间隙为4像素，这个问题产生的原因是换行引起的，因为我们写标签时通常会在标签结束符后顺手打个回车，而回车会产生回车符，回车符相当于空白符，通常情况下，多个连续的空白符会合并成一个空白符，而产生“空白间隙”的真正原因就是这个让我们并不怎么注意的空白符。

**b.**去除空隙的方法：  
　　1.对父元素添加，{font-size:0}，即将字体大小设为0，那么那个空白符也变成0px，从而消除空隙  
　　现在这种方法已经可以兼容各种浏览器，以前chrome浏览器是不兼容的  
图一：  


**c.**浏览器兼容性：ie6/7是不兼容 display：inline-block的所以要额外处理一下：  
　　在ie6/7下：  
　　对于行内元素直接使用{dislplay:inline-block;}  
　　对于块级元素：需添加{display:inline;zoom:1;}

**4.总结:**

　　display：inline-block的布局方式和浮动的布局方式，究竟使用哪个，我觉得应该根据实际情况来决定的：  
　　a.对于横向排列东西来说，我更倾向与使用inline-block来布局，因为这样清晰，也不用再像浮动那样清除浮动，害怕布局混乱等等。  
　　b.对于浮动布局就用于需要文字环绕的时候，毕竟这才是浮动真正的用武之地，水平排列的是就交给inline-block了。