z-index 详解

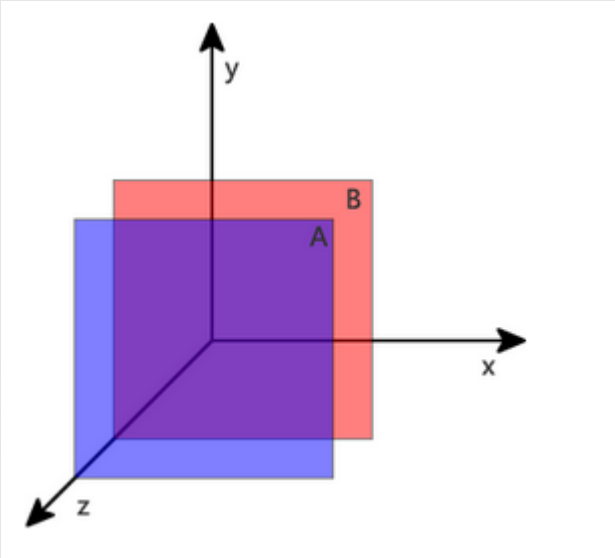
一、基本概念

z-index 属性设置元素的堆叠顺序。拥有更高堆叠顺序的元素总是会处于堆叠顺序较低的元素的前面。

注意：

1、元素可拥有负的 z-index 属性值。

2、Z-index 仅能在定位元素上奏效（例如 position:absolute;）。



说明：该属性设置一个定位元素沿 z 轴的位置，z 轴定义为垂直延伸到显示区的轴。如果为正数，则离用户更近，为负数则表示离用户更远。

二、可能的值

值	描述
auto	默认。堆叠顺序与父元素相等。
number	设置元素的堆叠顺序。
inherit	规定应该从父元素继承 z-index 属性的值。

三、层级关系的比较

1. 对于同级元素，默认(或 position:static)情况下文档流后面的元素会覆盖前面的。

2. 对于同级元素，position 不为 static 且 z-index 存在的情况下 z-index 大的元素会覆盖 z-index 小的元素，即 z-index 越大优先级越高。

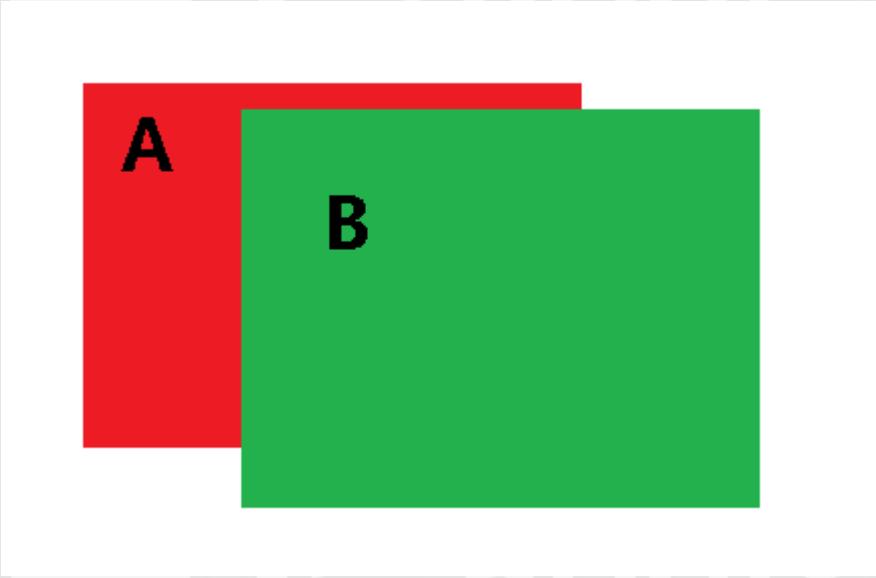
3. IE6/7 下 position 不为 static，且 z-index 不存在时 z-index 为 0，除此之外的浏览器 z-index 为 auto。

4. z-index 为 auto 的元素不参与层级关系的比较，由向上遍历至此且 z-index 不为 auto 的元素来参与比较。

四、顺序规则

如果不对节点设定 position 属性，位于文档流后面的节点会遮盖前面的节点。

```
<div id="a">A</div>
<div id="b">B</div>
```

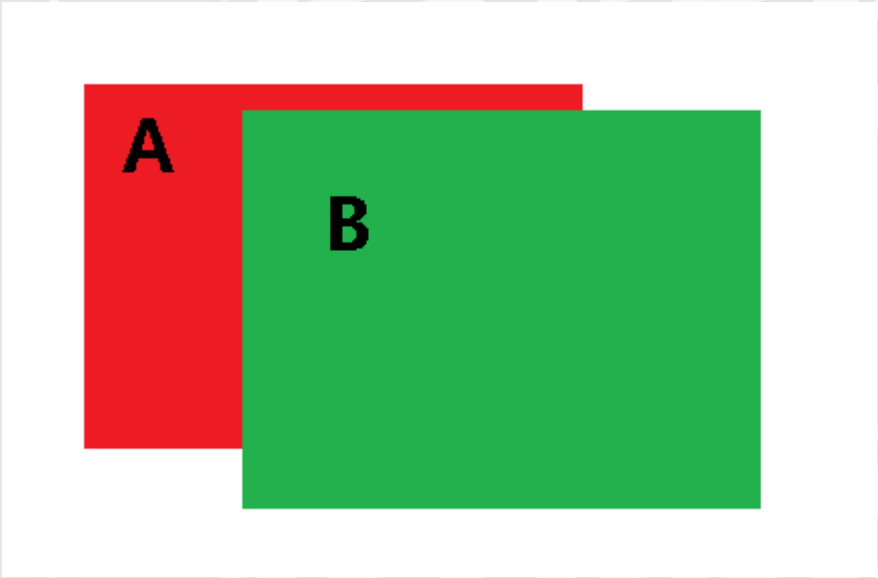


五、定位规则

（1）如果将 position 设为 static，位于文档流后面的节点依然会遮盖前面的节点浮

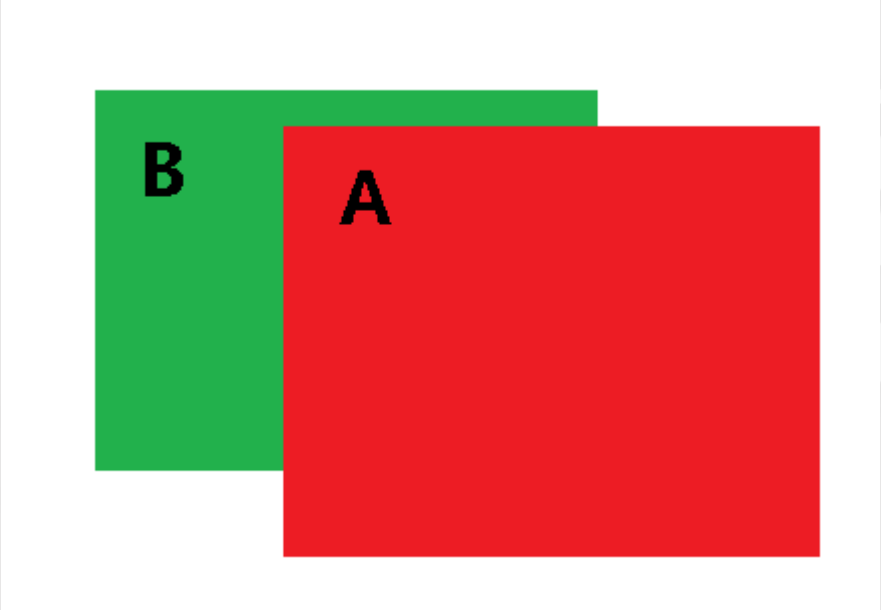
动,，所以 position:static 不会影响节点的遮盖关系。

```
<div id="a" style="position:static;">A</div>
<div id="b">B</div>
```



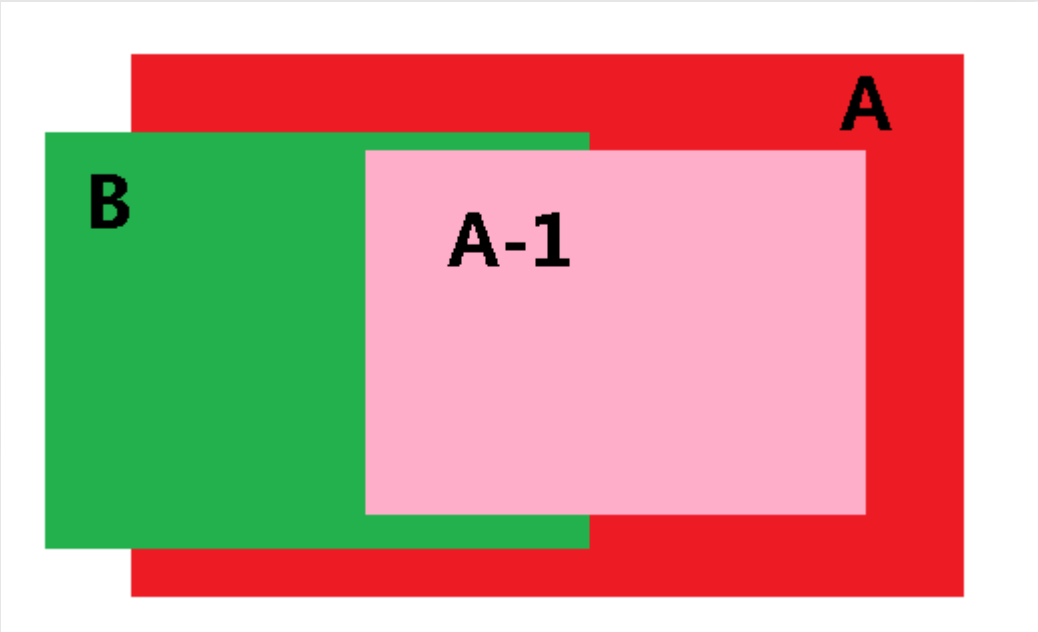
（2）如果将 position 设为 relative (相对定位)，absolute (绝对定位) 或者 fixed (固定定位)，这样的节点会覆盖没有设置 position 属性或者属性值为 static 的节点，说明前者比后者的默认层级高。

```
<div id="a" style="position:relative;">A</div>
<div id="b">B</div>
```



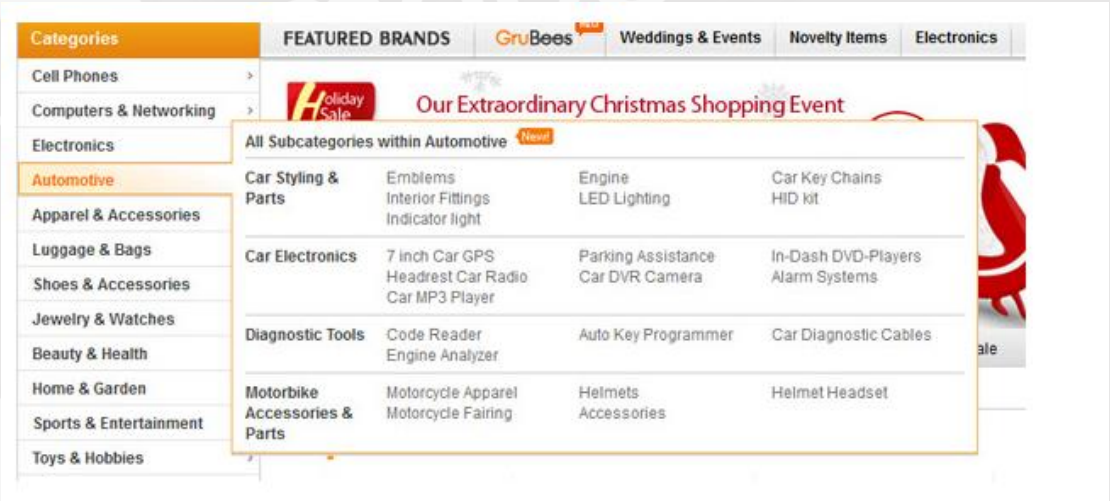
(3) 在没有 z-index 属性干扰的情况下，根据这顺序规则和定位规则，我们可以做出更加复杂的结构. 这里我们对 A 和 B 都不设定 position，但对 A 的子节点 A-1 设定 position:relative. 根据顺序规则，B 会覆盖 A，又根据定位规则 A' 会覆盖 B.

```
<div id="a">
    <div id="a-1" style="position:relative;">A-1</div>
</div>
<div id="b">B</div>
```



上面互相覆盖在什么时候用到这样的实现？看起来偏门，其实很常用，比如说，电子商务网站侧栏的类目展示列表就可以用这个技巧来实现.

下图是某网站的类目展示区域，二级类目的悬浮层覆盖一级类目列表外框，而一级类目的节点覆盖二级类目的悬浮层. 如果使用 CSS 实现展示效果，一级类目的外框相当于上面例子中的 A，一级类目的节点相当于 A-1，二级类目的悬浮层相当于 B.



六、参与规则

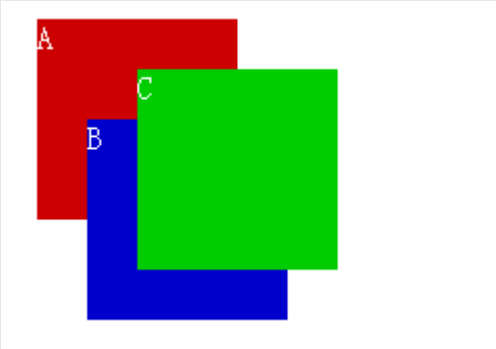
我们尝试不用 position 属性，但为节点加上 z-index 属性. 发现 z-index 对节点没起作用. z-index 属性仅在节点的 position 属性为 relative, absolute 或者 fixed 时生效.

例子一：

```
<div id="a" style="z-index:2;">A</div>

<div id="b" style="z-index:1;">B</div>

<div id="c" style="z-index:0;">C</div>
```

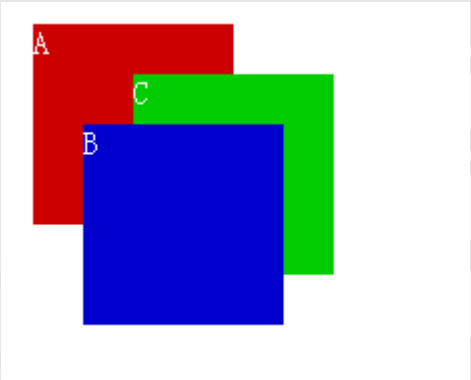


例子二：

```
<div id="a" style="z-index:2;">A</div>

<div id="b" style="position:relative;z-index:1;">B</div>

<div id="c" style="position:relative;z-index:0;">C</div>
```



七、默认值规则

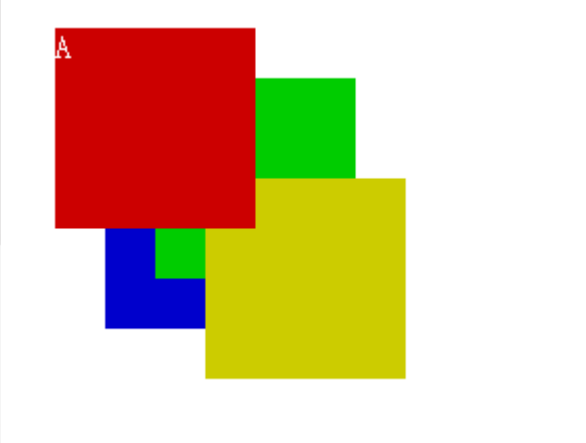
如果所有节点都定义了 position:relative. z-index 为 0 的节点与没有定义 z-index 在同一层级内没有高低之分；但 z-index 大于等于 1 的节点会遮盖没有定义 z-index 的节点; z-index 的值为负数的节点将被没有定义 z-index 的节点覆盖.

```
<div id="a" style="position:relative;z-index:1;">A</div>

<div id="b" style="position:relative;z-index:0;">B</div>

<div id="c" style="position:relative;">C</div>

<div id="d" style="position:relative;z-index:0;">D</div>
```

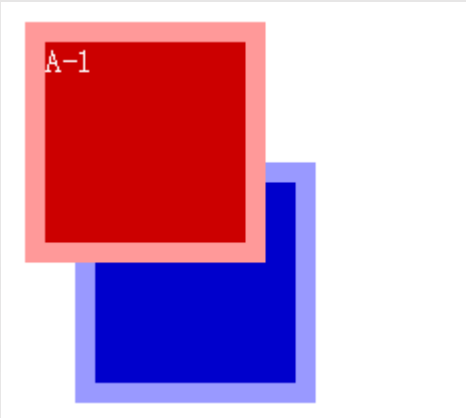


八、从父规则

如果 A, B 节点都定义了 position:relative, A 节点的 z-index 比 B 节点大, 那么 A 的子节点必定覆盖在 B 的子节点前面.

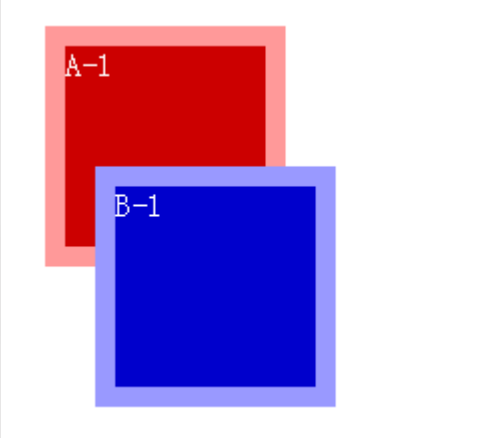
```
<div id="a" style="position:relative;z-index:1;">
    <div id="a-1">A-1</div>
</div>

<div id="b" style="position:relative;z-index:0;">
    <div id="b-1">B-1</div>
</div>
```



如果所有节点都定义了 position:relative, A 节点的 z-index 和 B 节点一样大,

但因为顺序规则，B 节点覆盖在 A 节点前面. 就算 A 的子节点 z-index 值比 B 的子节点大, B 的子节点还是会覆盖在 A 的子节点前面.



很多人将 z-index 设得很大，9999 什么的都出来了，如果不考虑父节点的影响，设得再大也没用，那是无法逾越的层级.