

# float

float是欺骗父元素，让其父元素误以为其高度塌陷了，但float元素本身仍处于文档流中，文字会环绕着float元素，不会被遮蔽。

# absolute

但absolute其实已经不能算是欺骗父元素了，而是出现了层级关系。

如果处于正常的文档流中的父元素算是凡人的话，那absolute已经得道成仙，用现在的话说已经不在一个次元上。

如图：从父元素的视点看，设成absolute的图片已经完全消失不见了，因此从最左边开始显示文字。而absolute的层级高，所以图片遮盖了文字。



```
<div style="border:4px solid blue;">
  
</div>
<div style="border:4px solid red;">
  
  我是一个绝对定位的absolute元素
</div>
```

```
<div style="border:4px solid blue;">
  
</div>
<div style="border:4px solid red;">
  
  我是一个绝对定位的absolute元素
```

</div>

通过对left/top/right/bottom的组合设置，absolute元素可以去任意它想去的地方，天空才是它的极限。

稍微补充一下：

- block元素通常被现实为独立的一块，会单独换一行；
- inline元素则前后不会产生换行，一系列inline元素都在一行内显示，直到该行排满。

## absolute和relative的应用

提示：

- 1.相对定位时，不必拘泥于relative+absolute，试试就去掉relative，充分利用absolute自身定位的特性，将relative和absolute解耦。耦合度越低维护起来越容易
- 2.拉伸平铺时，用relative可以有效限制止absolute元素的拉伸平铺范围（注意是拉伸，不是缩小。要缩小请再加上width/height:100%;）

## z-index的应用

敲黑板注意了：

以下情况根本不需要设z-index：

- 1.让absolute元素覆盖正常文档流内元素（不用设z-index，自然覆盖）
- 2.让后一个absolute元素覆盖前一个absolute元素（不用设z-index，只要在HTML端正确设置元素顺序即可）。  
如果你的页面不比京东更复杂，那z-index通常设成1,2,3足够了。

## 应用——使div分层

举个栗子:比如app图标右上角的消息通知



## HTML部分

```
<div style="display: inline-block;position:relative;margin-top:80px;">  
  <span class="icon">6</span>  
</div>
```

## CSS部分

```
.icon{  
  background-color: #f00;  
  color: #fff;  
  border-radius:50%;  
  text-align: center;  
  position: absolute;  
  width: 40px;  
  height: 40px;  
  right:-20px;  
  top:-20px;  
  font-size: 27px;  
}
```

最后效果：



6