不论是float还是position，使用他们都可以将行内元素转换为隐式的块级元素。

当容器的高度为auto，且容器中的元素浮动时，容器的高度不能自动伸长以适应内容的高度，使得内容溢出容器从而影响或破坏布局的现象叫浮动溢出。

|  |  |  |
| --- | --- | --- |
| 清除浮动的方法 | | |
| 在父盒子里面设置  overflow: hidden;  父盒子变成BFC，可以承载浮动元素 |  | 如果父盒子的高度小于子盒子，那么即使子盒子浮动，也还是会被裁剪 |
| 添加伪元素  .father : after {  content: “ ”;  height: 0;  line-height: 0;  display: block;  clear: both;  visibility: hidden; }  .father { **\***zoom: 1; } | 若父盒子的高度小于子盒子，这个方法会失效，所以无需为父盒子设置高度  **ch**i**ld**去看cvz（宠物展）  伪元素默认是行内元素 |
| 在子盒子后面加  <div class="clear"></div>  .clear {clear: both} | 会引入大量空标签，不推荐 |
| 在蓝色盒子上加  clear: both; | 2 | 引入大量空标签；  父盒子高度仍为0 |
| 伪元素是主要的清除浮动的方法，在小模块如ul中使用overflow: hidden | | |

浮动元素是被块级元素忽略的元素，但是行内元素和盒子里面的内容知道它们在哪里。嵌套在块级元素中的行内元素内容总会围绕着浮动元素，行内元素会留意浮动元素的边界，而块元素会正常流入页面。

|  |  |
| --- | --- |
| 无标题 | 无标题 |
|  | 为包含文字的div设置 {overflow: hidden;} |

浮动就像装死的教主，只有亲信（行内元素）知道他在哪里，还会守护在他周围，但是其他的人都当他没了。绝对定位就像是人死了，活着的人会填补空白继续生活。相对定位就习近平出访，虽然不在中南海，但是下边的人还是会以总书记为中心按部就班地生活。

|  |  |  |
| --- | --- | --- |
| <div>文字<img></div>，文字和<img>会排在一行  ① | img{ float: left; }  浮到左边之后仍然与文字一行  ② | img{ float: left;  margin-left: 30px;}  加margin后一行排不下了，所以换行 ③ |
| 11 |  | 11 |

对于一个div，padding, margin可以确定它的位置，但是float才能确定它的层级。从①到②，加了浮动之后，图片先浮起来，第三行文字就会跑上去，然后再浮向左边。③加了margin后，即使不浮动，也会换行排列。

脱离文档流的两种方法：浮动、绝对定位、固定定位。

一个大盒子中有一个小盒子，这时它们是在同一平面上的。如果要使两个盒子层叠就要使用定位。定位的时候，Z-index会被激活，被定位的元素就漂了起来。

相对定位不会脱离标准文档流，所以**层级并没有提高**，还会被后来流进来的元素覆盖。

Z-index的作用：设置两个定位过的元素的层次关系。两个原则：后来居上、拼爹。

但是在absolute / fixed时，如果不给父盒子设置定位，宽高都会以html（或上一级带定位的元素）为基准。

定位元素的**宽度**如果设置成100%，那么这个百分数针对的是参照物：

1. 如果是固定定位，不论有没有父盒子，它的宽度总是html的100%；
2. 绝对定位元素的宽度有三种情况：

|  |  |
| --- | --- |
| 父盒子未定位 | html的100% |
| 父盒子定位且设置了宽度px | 等于父盒子的宽度 |
| 父盒子相对定位但未设置宽度 | 等于父盒子的宽度 |
| 父盒子绝对定位但未设置宽度 | 重置为auto，即0 |

可见，未设置宽度的盒子让它绝对定位，它就没了宽度。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| position | static | absolute | | relative | | fixed |
| **参照物** |  | 若无父元素，相对于body定位；有父元素，但父元素无定位，仍相对于body定位；有父元素且父元素有定位，则相对于父元素定位。  （势利眼） | | 以box原来的位置为参照物发生偏移（想当年） | | browser的窗口window  （死心眼） |
| 滚动滚动条 |  | 随之滚动 | | 随之滚动 | | 不动 |
| 缩放窗口 |  | 不动 | | 不动 | | 不动 |
| 是否脱离文档流 |  | 脱离。  一个元素绝对定位时，会把它从页面的正常流中删除。也就是说绝对定位以后元素在页面中不占位置，对其他元素没有任何影响。  （人去楼空） | | 不脱离。  其他的元素（不论是行内还是块级）仍以原来的位置为基准正常注入。（停薪留职或者像是习大大出访了，但是下边该怎么运行还怎么运行） | | 脱离。  对其他元素没有任何影响。  （人去楼空） |
| 存在问题 |  | 随着浏览器窗口的大小而改变位置 | | 文字的基线和图片的底线对齐，文字会排在该定位元素之下 | |  |
| top: 8px; | right: 8px; | | bottom: 8px | left: 8px; | **tr**ou**bl**e | |

定位时写的Top / Left是以内盒子的**margin左上角**为基准，代表内（盒子）margin的外侧到外（盒子）border的内侧之间的距离。

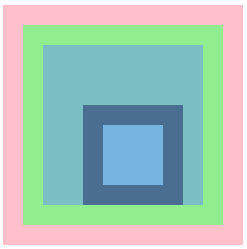
口诀：马外边内，边外内内。

定位 是从内盒子的 margin 外面到外盒子的 边框 里面；

margin是从内盒子的 边框 外面到外盒子的 内容框 里面。马仔把内裤穿外面

left: 50%; 是相对于**父**盒子宽度的一半。

例如/\*left: 0px;\*/属性是以边框对齐，如果不加的话，以内容框对齐。不论是浮动还是定位，margin都还在起作用。

小盒子： 宽高20px，内边距20px，边框20px，外边距20px，绝对定位left: 20px; top: 40px;

大盒子：宽高160px，内边距20px，边框20px;

|  |  |  |  |
| --- | --- | --- | --- |
|  | 初始宽度 | 父盒子浮动后，不管子盒子浮动与否，其宽度均由子盒子决定 | 父盒子定位后变成行内块级元素，其宽度由**未定位**的子元素撑开，要想让父元素独占一行，就要设置width: 100%； |
| 父盒子宽度 | 设置/继承上一级 |
| 子盒子宽度 | 设置/继承父盒子 |