

浮动

传统网页布局的三种方式

网页布局的本质——用css来摆放盒子。把盒子摆放到相应的位置。

css提供了三种传统布局方式（简单来说，就是盒子如何进行排列顺序）

- 普通流（标准流）
- 浮动
- 定位

标准流（普通流/文档流）

标签按照规定好的默认方式排列。

1、块级元素会独占一行，从上向下顺序排列。

常见元素：div、hr、p、h1-h6、ul、ol、dl、form、table等

2、行内元素会按照顺序，从左到右顺序排列，碰到父元素边缘自动换行。

常见元素：span、a、i、em等

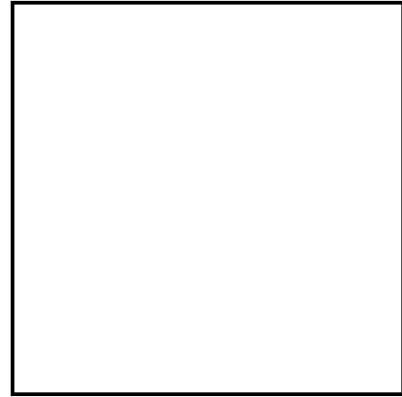
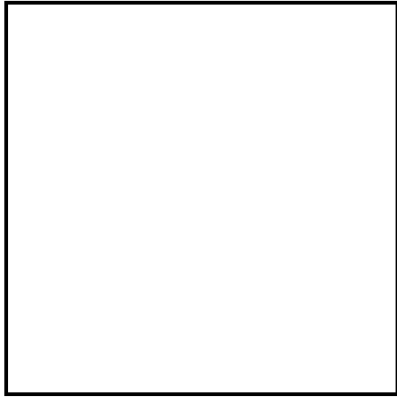
标准流是最基本的布局方式



```
1      <style type="text/css">
2          div{
3              width: 100px;
4              height: 100px;
5              border: 1px solid black;
6              display: inline-block;
7          }
8      </style>
9  </head>
10 <body>
11     <div></div>
12     <div></div>
13     <div></div>
14 </body>
```

为什么需要浮动

使用标准流无法实现左右对齐效果，有很多布局效果，标准流没有办法完成。此时可以利用浮动完成布局，因为浮动可以改变元素标签的默认方式。



```
1  <style type="text/css">
2      div{
3          width: 100px;
4          height: 100px;
5          border: 1px solid black;
6      }
7      .a{
8          float: right;
9      }
10 </style>
11 </head>
12 <body>
13     <div class="a"></div>
14     <div class="b"></div>
15 </body>
```

浮动最典型的应用：可以让多个块级元素一行内排列显示。

网页第一准则：多个块级元素纵向排列标准流，多个块级元素横向排列找浮动。

浮动的属性及属性值

语法：

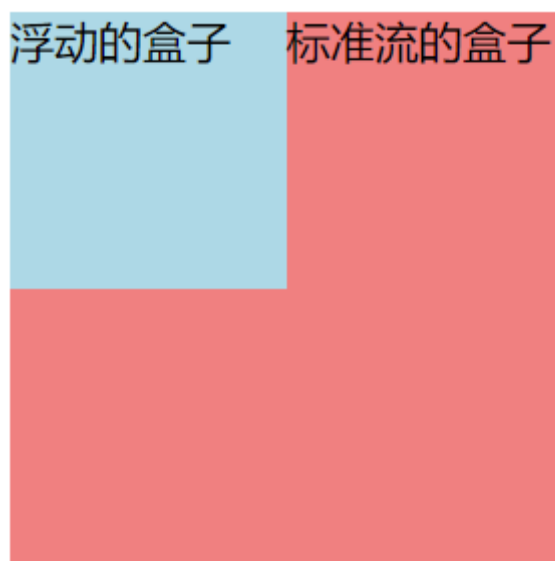
```
1  .a{
2      float: left;
3  }
4
5  选择器{
6      float: 属性值;
7  }
```

属性值	描述
none	元素不浮动（默认值）
left	元素向左浮动
right	元素向右浮动

浮动的特性

设置了浮动float的元素最重要的特性

- 1、脱离标准普通流的控制，移动到指定位置。
- 2、浮动的盒子不再保留原先的位置。



```
1      <style type="text/css">
2          .a{
3              width: 100px;
4              height: 100px;
5              background: lightblue;
6              float: left;
7          }
8          .b{
9              width: 200px;
10             height: 200px;
11             background: lightcoral;
12         }
13     </style>
14 </head>
15 <body>
16     <div class="a">浮动的盒子</div>
17     <div class="b">标准流的盒子</div>
18 </body>
```

3、浮动元素会具有行内块元素的特性。（任何元素都可以设置浮动属性，行内元素设置浮动后，也会具有行内块元素的特性。）

例子：使用span标签添加浮动属性，展现span标签具有行内块的效果。



```
1      <style type="text/css">
2          span{
3              float: left;
4              width: 200px;
5              height: 100px;
6              background: red;
7          }
8      </style>
9  </head>
10 <body>
11     <span>1</span>
12     <span>2</span>
13 </body>
```

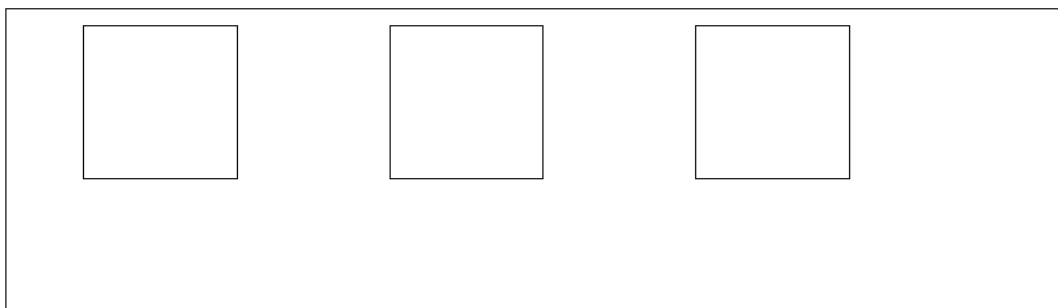
注意：如果行内元素有了浮动，则不需要转换块级/行内元素就可以直接给高度和宽度。

4、浮动的盒子中间是没有缝隙的，是紧挨着一起的。

浮动元素经常和标准流父级搭配使用

为了约束浮动元素的位置，我们网页布局一班采取的策略是：

先用标准流的父元素排列上下位置，之后内部子元素采取浮动排列左右位置。



```
1      <style type="text/css">
2          .box{
3              width: 700px;
4              height: 200px;
5              border: 1px solid black;
```

```

6         margin: 0 auto;
7     }
8     .box_a{
9         width: 100px;
10        height: 100px;
11        border: 1px solid black;
12        float: left;
13        margin: 10px 50px;
14    }
15    </style>
16 </head>
17 <body>
18     <div class="box">
19         <div class="box_a"></div>
20         <div class="box_a"></div>
21         <div class="box_a"></div>
22     </div>
23 </body>

```

案例: <https://www.mi.com/index.html>



一个元素浮动后，理论上其他的兄弟元素也要浮动

一个盒子里面有多个盒子，如果其中一个盒子浮动后，那么其他兄弟元素也应该浮动，以防止引起布局变乱的问题。

浮动的盒子只会影响盒子后面的标准流，不会影响前面的标准流。



```

1     <style type="text/css">
2         .box{
3             width: 700px;

```

```
4         height: 200px;
5         border: 1px solid black;
6         margin: 0 auto;
7     }
8     .box_a1{
9         width: 100px;
10        height: 100px;
11        border: 1px solid black;
12    }
13    .box_a2{
14        width: 130px;
15        height: 130px;
16        border: 1px solid black;
17        float: left;
18    }
19    .box_a3{
20        width: 150px;
21        height: 150px;
22        border: 1px solid black;
23    }
24    </style>
25 </head>
26 <body>
27     <div class="box">
28         <div class="box_a1 fl">盒子1</div>
29         <div class="box_a2">盒子2</div>
30         <div class="box_a3">盒子3</div>
31     </div>
32 </body>
```

清除浮动

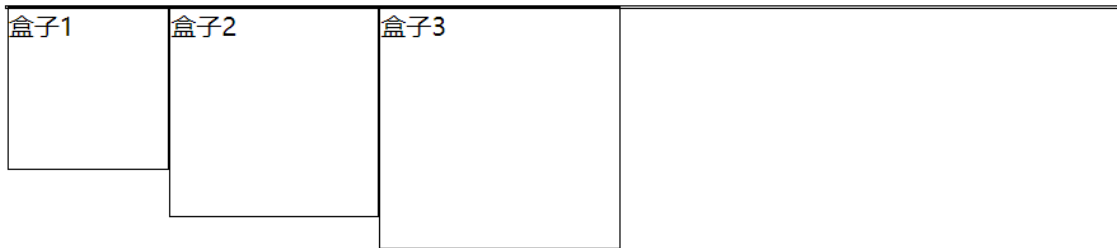
为什么要清除浮动

之前浮动元素有一个标准流的父元素，都用一个共同点，都是有设置高度的。

正常网页开发中，对于父元素的div的高度大部分情况下，高度都是根据子元素的内容撑开的。

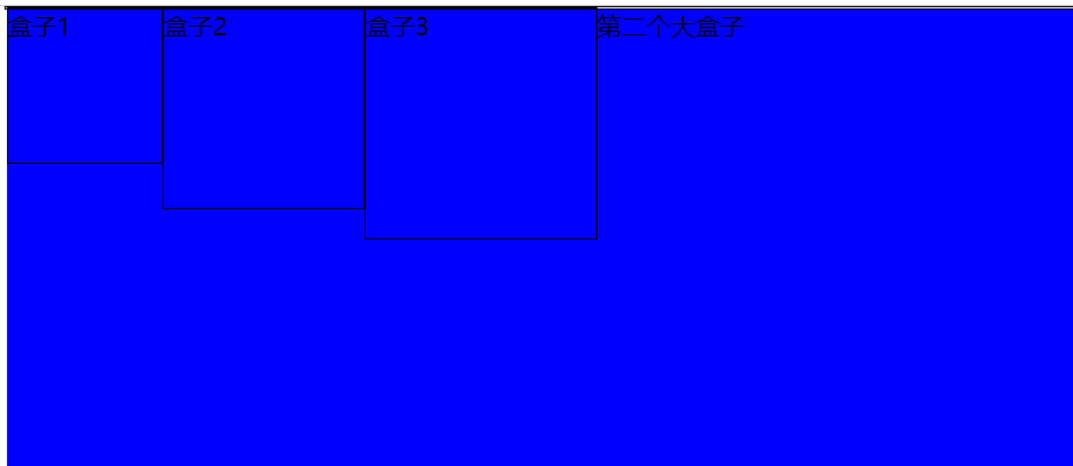
但是，如果父元素不设置高度的情况下，其子元素进行浮动后。父元素的高度因为子元素进行浮动脱离了文档流。所以父元素的高度在标准流里默认成了0px。

就会产生如下效果。（父元素的高度成了0px）



```
1      <style type="text/css">
2          .box{
3              width: 700px;
4              border: 1px solid black;
5              margin: 0 auto;
6          }
7          .box_a1{
8              width: 100px;
9              height: 100px;
10             border: 1px solid black;
11             float: left;
12         }
13         .box_a2{
14             width: 130px;
15             height: 130px;
16             border: 1px solid black;
17             float: left;
18         }
19         .box_a3{
20             width: 150px;
21             height: 150px;
22             border: 1px solid black;
23             float: left;
24         }
25     </style>
26 </head>
27 <body>
28     <div class="box">
29         <div class="box_a1 fl">盒子1</div>
30         <div class="box_a2">盒子2</div>
31         <div class="box_a3">盒子3</div>
32     </div>
33 </body>
```

如果在第一个高度为0的盒子后面再加入其他盒子的话，就会产生如下效果。严重影响网页布局的效果。



```
1      <style type="text/css">
2          *{
3              margin: 0 auto;
4          }
5          .box{
6              width: 700px;
7              border: 1px solid black;
8          }
9          .box_a1{
10             width: 100px;
11             height: 100px;
12             border: 1px solid black;
13             float: left;
14         }
15         .box_a2{
16             width: 130px;
17             height: 130px;
18             border: 1px solid black;
19             float: left;
20         }
21         .box_a3{
22             width: 150px;
23             height: 150px;
24             border: 1px solid black;
25             float: left;
26         }
27         .box2{
28             width: 700px;
29             height: 300px;
30             background: blue;
31         }
32     </style>
33 </head>
34 <body>
35     <div class="box">
36         <div class="box_a1 fl">盒子1</div>
37         <div class="box_a2">盒子2</div>
38         <div class="box_a3">盒子3</div>
39     </div>
40     <div class="box2">
41         第二个大盒子
```



```
42     </div>
43 </body>
```

如何清除浮动

清除浮动的本质：清除浮动元素造成的影响。

语法：

```
1  选择器{
2      clear: 属性值;
3  }
```

属性值	描述
left	不允许左侧有浮动元素（清除左侧浮动的影响）
right	不允许右侧有浮动元素（清除右侧浮动的影响）
both（使用率最高）	同时清除左右两侧浮动的影响

清除浮动的方法

1、清除浮动——额外标签法

额外标签法也称为隔墙法，是常用的推荐做法。

额外标签法会在浮动元素末尾添加一个空的标签（空标签必需是块级元素。）。例如：`<div style="clear: both;"></div>`，或者其他标签（标签必需是块级元素）

优点：通俗易懂，书写方便。

缺点：添加许多无意义的标签，容易影响标签结构观感。

```
1      <style type="text/css">
2          *{
3              margin: 0 auto;
4          }
5          .box{
6              width: 700px;
7              border: 1px solid black;
8          }
9          .box_a1{
10             width: 100px;
11             height: 100px;
12             border: 1px solid black;
13             float: left;
14         }
15         .box_a2{
16             width: 130px;
17             height: 130px;
18             border: 1px solid black;
19             float: left;
20         }
```

```

21     .box_a3{
22         width: 150px;
23         height: 150px;
24         border: 1px solid black;
25         float: left;
26     }
27     .box2{
28         width: 700px;
29         height: 300px;
30         background: blue;
31     }
32 </style>
33 </head>
34 <body>
35     <div class="box">
36         <div class="box_a1 fl">盒子1</div>
37         <div class="box_a2">盒子2</div>
38         <div class="box_a3">盒子3</div>
39         <div style="clear: both;"></div>
40     </div>
41     <div class="box2">
42         第二个大盒子
43     </div>
44 </body>

```

2.清除浮动——给父级添加overflow

语法:

```

1  选择器{
2      overflow: 属性值;
3  }

```

给**父级**添加overflow属性, 将其属性值设置为hidden、auto、或者scroll

优点: 代码简洁, 不像第一种方法, 不影响网页代码标签观感。

缺点: 无法显示溢出部分。(后期会对此属性进行详细讲解。)

3.清除浮动——:after伪元素法

:after方式是额外标签法的升级版, 也是给**父元素**添加, 不过是添加css属性并非标签。

```

1  .clearfix::after{
2      content: "";
3      display: block;
4      height: 0;
5      clear: both;
6      visibility: hidden;
7  }
8  .clearfix{
9      /* IE6、7 */
10     *zoom: 1;
11 }

```

优点：没有增加标签，结构更简单

缺点：照顾低版本浏览器，增加代码量

清除浮动总结

为什么要清除浮动

- 1、父级没高度
- 2、子盒子浮动后。
- 3、影响下面元素的布局，需要清除浮动，保持页面整洁。

清除浮动的方式	优点	缺点
额外标签法	通俗易懂，书写方便	添加许多无意义的标签，结构化较差
父级overflow: hidden	书写简单	溢出隐藏
父级after伪元素	结构语义化正确	由于IE6-7不支持：after，兼容性问题。