

# 圣杯布局实现思路

根据课程中的讲解，圣杯布局代码以及效果如下：

```
8 * {
9     margin: 0;
10    padding: 0;
11 }
12 body {
13     min-width: 700px;
14 }
15 .header,
16 .footer {
17     float: left;
18     width: 100%;
19     background: #ddd;
20     height: 40px;
21     line-height: 40px;
22     text-align: center;
23 }
24 .container {
25     padding: 0 220px 0 200px;
26 }
27 .left,
28 .middle,
29 .right {
30     position: relative;
31     float: left;
32     min-height: 300px;
33 }
34
35 .left {
36     margin-left: -100%;
37     left: -200px;
38     width: 200px;
39     background: #f00;
40 }
41
42 .right {
43     margin-left: -220px;
44     right: -220px;
45     width: 220px;
46     background: #30a457;
47 }
48
49 .middle {
50     width: 100%;
51     background: #1a5acd;
52 }
```

```
<div class="header">
  <h4>header</h4>
</div>
<div class="container">
  <div class="middle">
    <h4>middle</h4>
    <p>
      这是页面的主体内容 这是页面的主体内容 这是页面的主体内
      容 这是页面的主体内容
    </p>
  </div>
  <div class="left">
    <h4>left</h4>
    <p>
      这是页面的左边 这是页面的左边 这是页面的左边 这是页面
    </p>
  </div>
  <div class="right">
    <h4>right</h4>
    <p>
      这是页面的右边 这是页面的右边 这是页面的右边 这是页面
    </p>
  </div>
</div>
<div class="footer">
  <h4>footer</h4>
</div>
```

## 一、左边块如何移上去，到达最左边

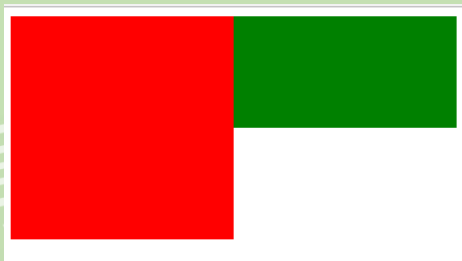
1、要了解这个，首先需要了解 margin-left 设置为负值时，元素会往左移。例：

(1) 首先是正常显示时的效果，.red 与.green 呈水平排列。

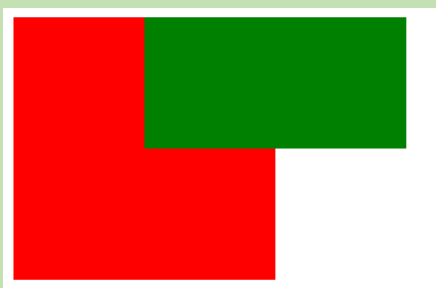
```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style type="text/css">
    .red{
      width: 200px;
      height: 200px;
      background-color: red;
      float: left;
    }
    .green{
      width: 200px;
      height: 100px;
      background-color: green;
      float: left;
    }
  </style>
</head>
<body>
  <div class="red"></div>
  <div class="green"></div>
</body>
</html>

```



- (2) 然后给第二个块.green 添加 margin-left: -100px;属性的时候，第二个块会向左移动 100px，与第一个重合了，可观察如下效果图。



- 2、实现圣杯布局时，.left 元素与.right 元素在没有设置 margin-left、与 left，right 属性时。左边与右边都没有上移的效果，如下代码与效果：

**注意：**这里本来这三个块应该在一行中显示的（因为有添加浮动），然后由于中间的块 middle 的宽度设置为 100%，所以把另外两个块给挤下去了（左右有两块空白，是由于父级设置了 padding 属性）。

```

.left {
  /*margin-left: -100%;*/
  /*left: -200px;*/
  width: 200px;
  background: #f00;
}
.right {
  /*margin-left: -220px;*/
  /*right: -220px;*/
  width: 220px;
  background: #30a457;
}
middle {

```

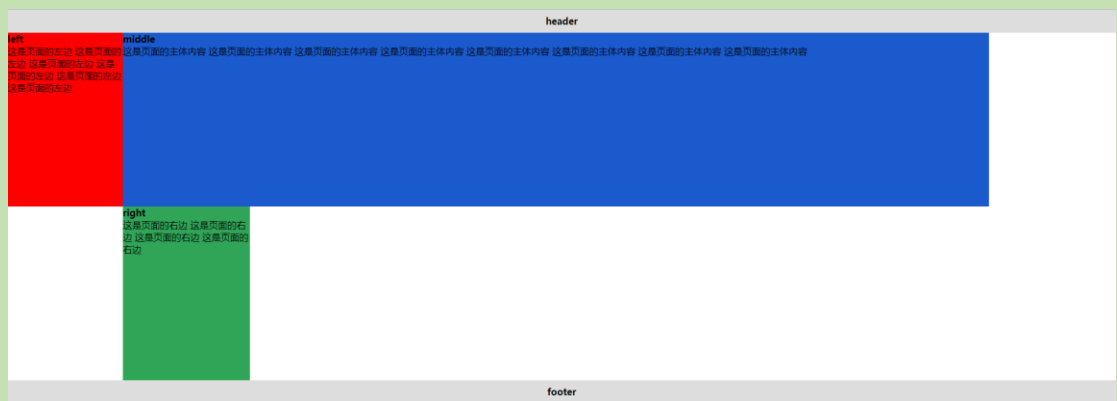


- (1) 根据上面了解到的情况,所以当.left元素设置 margin-left: -100%;属性的时候,.left这个块会往左移动 100%,这里的 100%是父级的宽度的 100%,就正好到了父级的左边(.middle的宽度也是 100%,正好充满父级的 width,不包括 padding,所以也到了.middle元素的左边)。效果如下图。

```
.left {
  margin-left: -100%;
  /*left: -200px;*/
  width: 200px;
  background: #f00;
}
```



- (2) 同理,添加 left: -200px;属性,又向左移动了 200px,200px 正好是自身的宽度,所以整个移到了浏览器窗口的最左边。



(3) 同理，.right 元素设置 margin-left: -220px;与 right: -220px;属性，右边的块到达最右边。



## 二、关于最小值的设置

1、设置整体的最小值需要满足  $\text{min-width} \geq 2X + Y$ ;其中 X 指左边块的宽度，Y 为右边块的宽度，所以最小值需要大于等于两个左边块与一个右边块的宽度之和。等于的时候，是临界值。