

# 解决父元素高度坍塌问题

子元素使用了float之后，父级盒子的高度会变为0，称为父元素高度坍塌。

## 一、问题形成

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>清除浮动</title>
  <style>
    * {
      margin: 0;
      padding: 0;
    }

    .outer {
      width: 80vw;
      margin: 0 auto;
      background-color: #2a6288;
      padding: 10px;
    }

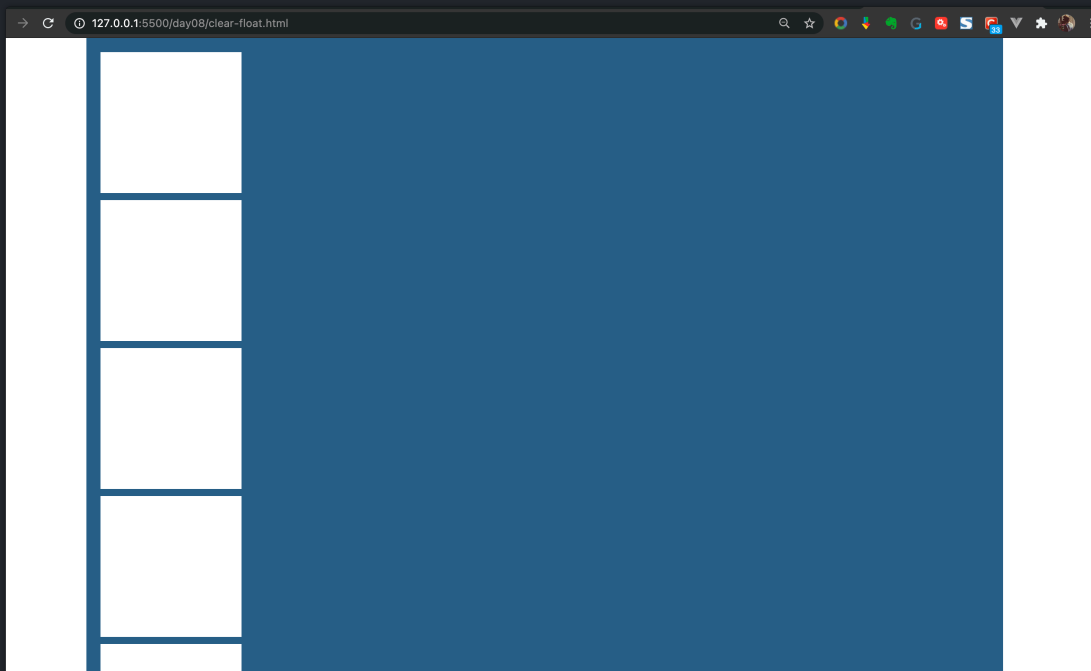
    .outer .inner {
      width: 200px;
      height: 200px;
      background-color: #fff;
      margin: 10px;
    }
  </style>
</head>
```

```
<body>
  <div class="outer">
    <div class="inner"></div>
    <div class="inner"></div>
    <div class="inner"></div>
    <div class="inner"></div>
    <div class="inner"></div>
  </div>

</body>

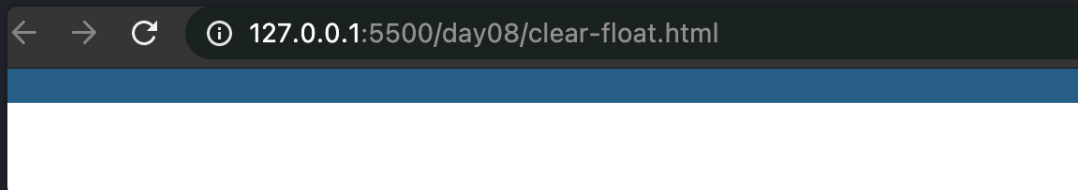
</html>
```

## 运行效果



当我们给每个内部的div加上左浮动属性后，它们将从文档流脱离，又因为父元素没有指定高度，所以导致父元素无法获得高度，形成如下图所示的高度坍塌问题

```
.outer .inner {
  width: 200px;
  height: 200px;
  background-color: #fff;
  margin: 10px;
  float: left;
}
```

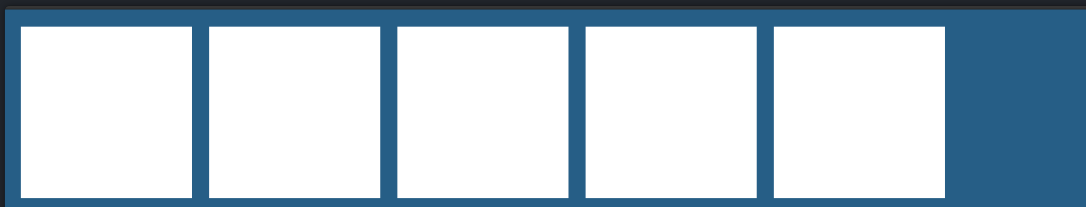


## 二、解决方法

### 1、给父级元素单独定义高度（height）

```
.outer {
  width: 80vw;
  margin: 0 auto;
  background-color: #2a6288;
  padding: 10px;
  /* 指定高度 */
  height: 220px;
}
```

效果



**原理：**如果父级元素没有定义高度，父元素的高度完全由子元素撑开时，父级div手动定义height，就解决了父级div无法自动获取到高度的问题。

**优点：**简单、代码少、容易掌握。

**缺点：** 只适合高度固定的布局，要给出精确的高度，如果高度和父级div不一样时，会产生问题。对于响应式布局会有很大影响。

## 2、在标签结尾处加空div标签 clear:both

不给父元素设置高度，在最后加上清除浮动的空div

```
15  .outer {  
16      width: 80vw;  
17      margin: 0 auto;  
18      background-color: #2a6288;  
19      padding: 10px;  
20      /* height: 220px; */  
21  }  
22  
23  .outer .inner {  
24      width: 200px;  
25      height: 200px;  
26      background-color: #fff;  
27      margin: 10px;  
28      float: left;  
29  }  
30  
31  .clear {  
32      clear: both;  
33  }  
34  </style>  
35  </head>  
36  
37  <body>  
38      <div class="outer">  
39          <div class="inner"></div>  
40          <div class="inner"></div>  
41          <div class="inner"></div>  
42          <div class="inner"></div>  
43          <div class="inner"></div>  
44          <div class="clear"></div>  
45      </div>  
46  
47  </body>
```

父元素不指定高度

效果同上

**原理：** 添加一个空div，利用css提高的clear:both清除浮动，让父级div能自动获取到高度。

**优点：** 简单、代码少、浏览器支持好、不容易出现怪问题

**缺点：**不少初学者不理解原理；如果页面浮动布局多，就要增加很多空div，不利于页面的优化。

### 3、父级div定义 伪类:after 和 zoom

```
34
35     .clearfix {
36         zoom: 1;
37     }
38
39     .clearfix:after {
40         display: block;
41         clear: both;
42         content: "";
43         visibility: hidden;
44         height: 0;
45     }
46 </style>
47 </head>
48
49 <body>
50     <div class="outer clearfix">
51         <div class="inner"></div>
52         <div class="inner"></div>
53         <div class="inner"></div>
54         <div class="inner"></div>
55         <div class="inner"></div>
56         <!-- <div class="clear"></div> -->
57     </div>
58
```

添加 clearfix 样式

**原理：**元素生成伪类的作用和效果相当于方法2中的原理，但是IE8以上和非IE浏览器才支持:after，zoom可解决ie6,ie7浮动问题

**优点：**浏览器支持好、不容易出现奇怪问题，写法是固定的，不理解也可以直接复制使用；（推荐使用此种方法，简单便捷，只需添加一个class即可解决问题）

**缺点：**css代码多、不少初学者不理解原理，要两句代码结合使用才能让主流浏览器都支持。

### 4、父级div定义 overflow:hidden

```
.outer {
  width: 80vw;
  margin: 0 auto;
  /* height: 220px; */
  background-color: #2a6288;
  padding: 10px;
  overflow: hidden;
}
```

```
<div class="outer">
  <div class="inner"></div>
  <div class="inner"></div>
  <div class="inner"></div>
  <div class="inner"></div>
  <div class="inner"></div>
  <!-- <div class="clear"></div> -->
</div>
```

**优点：**简单、代码少、浏览器支持好

**缺点：**不能和position配合使用，因为超出的尺寸的会被隐藏。（不建议使用此种方式，可能会影响页面元素布局）

```
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32

.outer {
  width: 80vw;
  margin: 0 auto;
  /* height: 220px; */
  background-color: #2a6288;
  padding: 10px;
  overflow: hidden;
}

.outer .inner {
  width: 200px;
  height: 200px;
  background-color: #fff;
  float: left;
  margin: 10px;
  position: absolute;
}
```

父元素设置为 overflow:hidden 的缺点

设置 position 后，父元素高度又坍塌了

## 5、父级div定义 overflow:auto

```
.outer {
  width: 80vw;
  margin: 0 auto;
  background-color: #2a6288;
  padding: 10px;
  /* height: 220px; */
  overflow: auto;
}
```

```
<body>
  <div class="outer">
    <div class="inner"></div>
    <div class="inner"></div>
    <div class="inner"></div>
    <div class="inner"></div>
    <div class="inner"></div>
    <!-- <div class="clear"></div> -->
  </div>
```

**原理：**必须定义width或zoom:1，同时不能定义height，使用overflow:auto时，浏览器会自动检查浮动区域的高度

**优点：**简单、代码少、浏览器支持好

**缺点：**内部宽高超过父级div时，会出现滚动条。