

第7章 CSS的布局模型

笔记本: CSS

创建时间: 6/4/2019 3:51 AM

更新时间: 5/18/2020 8:26 PM

作者: tanziqu1756@outlook.com

URL: <https://www.imooc.com/code/2073>

在网页中，元素有三种布局模型：

- 1、流动模型 (Flow)**
- 2、浮动模型 (Float)**
- 3、层模型 (Layer)**

一、流动模型（默认的网页布局模式）

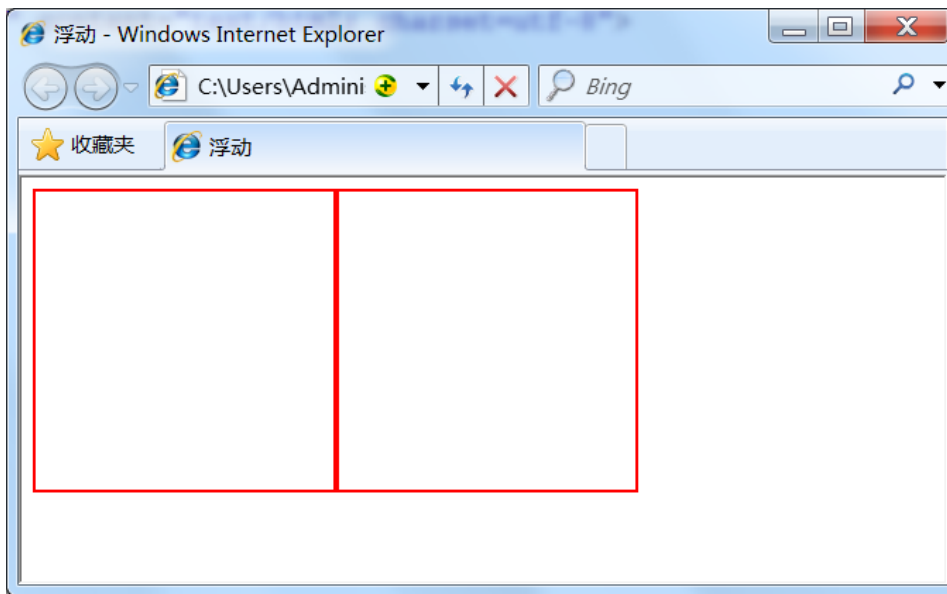
特征：

- 1、块状元素都会在所处的包含元素内自上而下按顺序垂直延伸分布，
因为在默认状态下，块状元素的宽度都为100%。实际上，块状元素都会以行的形式占据位置。**
- 2、在流动模型下，内联元素都会在所处的包含元素内从左到右水平分布显示。**

如果想让块状元素浮动显示，也是可以的！

如下代码，让两个div并排出现在同一行。

```
div{  
    width:200px;  
    height:200px;  
    border:2px red solid;  
    float:left;  
}
```



请注意，添加了浮动float属性的盒子，会浮动于没有设置浮动属性的盒子的上方。

二、层模型

由于网页大小的活动性，层布局没能受到热捧。

但是在网页上局部使用层布局还是有其方便之处的。

CSS定义了一组定位（positioning）属性来支持层布局模型。

层模型有三种形式：

1、绝对定位(position: absolute)：这条语句的作用将元素从文档流中拖出来，然后使用left、right、top、bottom属性相对于其最接近的一个具有定位属性的父包含块进行绝对定位。如果不存在这样的包含块，则相对于body元素，即相对于浏览器窗口。

2、相对定位(position: relative)相对定位完成的过程是首先按static(float)方式生成一个元素(并且元素像层一样浮动了起来)，然后相对于以前的位置移动，移动的方向和幅度由left、right、top、bottom属性确定，偏移前的位置保留不动。

3、固定定位(position: fixed)

2.1 绝对定位

以浏览器为绝对静止参考系。

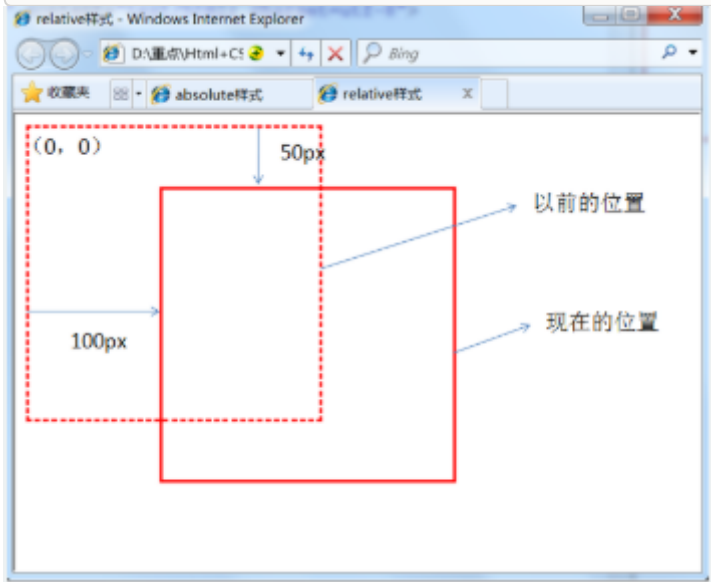
```
div{
    width:200px;
    height:200px;
    border:2px red solid;
    position:absolute;
    left:100px;
```

```
top:50px;
}
<div id="div1"></div>
```

2.2相对定位

相对于默认位置的移动。

```
position: relative;
```



```
index.html
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>relative样式</title>
7   <style type="text/css">
8     #div1 {
9       width: 200px;
10      height: 200px;
11      border: 2px red solid;
12      position: relative;
13      left: 100px;
14      top: 50px;
15    }
16  </style>
17 </head>
18
19 <body>
20   <div id="div1"></div>
21   <span>偏移前的位置还保留不动，覆盖不了前面的div没有偏移前的位置</span>
22 </body>
23
24 </html>
```

偏移前的位置还保留不动，覆盖不了前面的div没有偏移前的位置

2.3固定定位

不管如何拖动滚动条，其位置都不会变。就像置顶了一样。

三、相对定位与绝对定位的配合使用

请注意谁是绝对定位，谁是相对定位。

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
<title>相对参照元素进行定位</title>
```

```

<style type="text/css">
div{border:2px red solid;}
#box1{
    width:200px;
    height:200px;
    position:relative;
}
#box2{
    position:absolute;
    top:20px;
    left:30px;

}
/*下面是任务部分*/
#box3{
    width:200px;
    height:200px;
    position:relative;
}
#box4{
    width:99%;
    position:absolute;
    bottom:0;
}
</style>
</head>

<body>
<div id="box1">
    <div id="box2">相对参照元素进行定位</div>
</div>

<h1>下面是任务部分</h1>
<div id="box3">
    
    <div id="box4">当我还是三年级的学生时是一个害羞的小
女生。</div>
</div>
</body>
</html>

```

注意：小div是根据大div进行绝对定位的。

相对参照元素进行定位

下面是任务部分



四、flex弹性盒模型

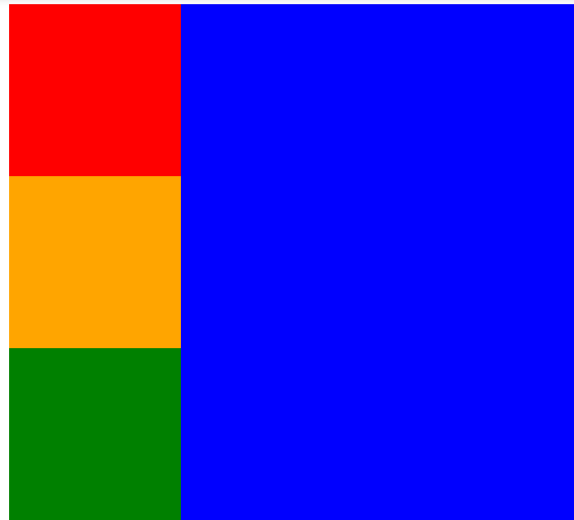
技术点的解释：

- 1、设置display: flex属性可以把块级元素在一排显示。
- 2、flex需要添加在父元素上，改变子元素的排列顺序。
- 3、默认为从左往右依次排列,且和父元素左边没有间隙。

```

index.html
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>flex布局</title>
7   <style type="text/css">
8     .box {
9       background: blue;
10      /*display: flex;*/
11    }
12
13    .box div {
14      width: 200px;
15      height: 200px;
16    }
17
18    .box1 {
19      background: red;
20    }
21
22    .box2 {
23      background: orange;
24    }
25
26    .box3 {
27      background: green;
28    }
29  </style>
30 </head>
31
32 <body>
33   <div class="box">
34     <div class="box1"></div>
35     <div class="box2"></div>
36     <div class="box3"></div>
37   </div>
38 </body>
39
40 </html>
41

```



```

index.html
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>flex布局</title>
7   <style type="text/css">
8     .box {
9       background: blue;
10      display: flex;
11    }
12
13    .box div {
14      width: 200px;
15      height: 200px;
16    }
17
18    .box1 {
19      background: red;
20    }
21
22    .box2 {
23      background: orange;
24    }
25
26    .box3 {
27      background: green;
28    }
29  </style>
30 </head>
31
32 <body>
33   <div class="box">
34     <div class="box1"></div>
35     <div class="box2"></div>
36     <div class="box3"></div>
37   </div>
38 </body>
39
40 </html>
41

```



使用justify-content属性设置横轴排列方式



`flex-end` : 右对齐

```
.box {  
  background: blue;  
  display: flex;  
  justify-content: flex-end;  
}
```

实现效果:



`center` : 居中

```
.box {  
  background: blue;  
  display: flex;  
  justify-content: center;  
}
```

实现效果:



`space-between` : 两端对齐, 项目之间的间隔都相等。

```
.box {  
  background: blue;  
  display: flex;  
  justify-content: space-between;  
}
```

实现效果:



```

index.html
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>justify-content</title>
7   <style type="text/css">
8     .box {
9       background: blue;
10      display: flex;
11      justify-content: space-between;
12      /*justify-content: flex-start | flex-end | center |
13         space-between | space-around;*/
14    }
15
16    .box div {
17      width: 200px;
18      height: 200px;
19    }
20
21    .box1 {
22      background: red;
23    }
24
25    .box2 {
26      background: orange;
27    }
28
29    .box3 {
30      background: green;
31    }
32  </style>
33 </head>
34
35 <body>
36   <div class="box">
37     <div class="box1"></div>
38     <div class="box2"></div>
39     <div class="box3"></div>
40   </div>
41 </body>
42

```

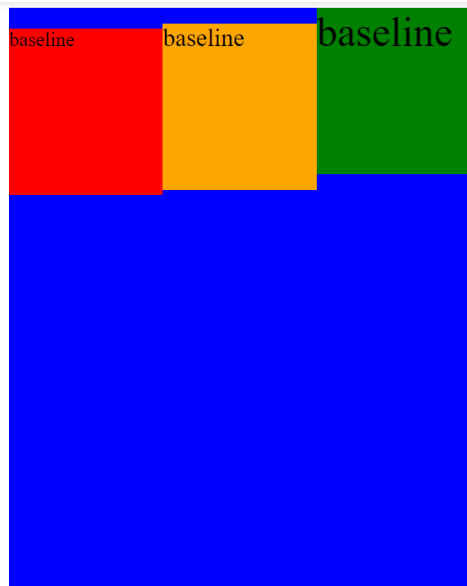


使用align-items属性设置纵轴排列方式

```

index.html
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>align-items</title>
7   <style type="text/css">
8     .box {
9       height: 700px;
10      background: blue;
11      display: flex;
12      align-items: baseline;
13      /* align-items: flex-start | flex-end | center | baseline | stretch; */
14      /* baseline: 项目的第一行文字的基线对齐 */
15      /* stretch: (默认值)：如果项目未设置高度或设为auto，将占满整个容器的高度。 */
16    }
17
18    .box div {
19      width: 200px;
20      height: 200px;
21    }
22
23    .box1 {
24      background: red;
25    }
26
27    .box2 {
28      font-size: 30px;
29      background: orange;
30    }
31
32    .box3 {
33      font-size: 50px;
34      background: green;
35    }
36  </style>
37 </head>
38
39 <body>
40   <div class="box">
41     <div class="box1">baseline</div>
42     <div class="box2">baseline</div>
43     <div class="box3">baseline</div>
44   </div>
45

```



给子元素设置flex占比

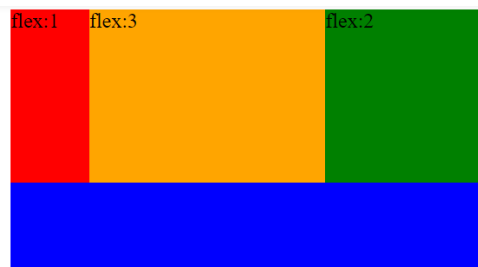
技术点的解释：

- 1、给子元素设置flex属性,可以设置子元素相对于父元素的占比。
- 2、flex属性的值只能是正整数,表示占比多少。
- 3、给子元素设置了flex之后,其宽度属性会失效。


```

index.html
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>flex占比</title>
7   <style type="text/css">
8     .box {
9       height: 300px;
10      background: blue;
11      display: flex;
12    }
13
14    .box div {
15      width: 200px;
16      height: 200px;
17    }
18
19    .box1 {
20      flex: 1;
21      background: red;
22    }
23
24    .box2 {
25      flex: 3;
26      background: orange;
27    }
28
29    .box3 {
30      flex: 2;
31      background: green;
32    }
33  </style>
34 </head>
35
36 <body>
37   <div class="box">
38     <div class="box1"></div>
39     <div class="box2"></div>
40     <div class="box3"></div>
41   </div>
42 </body>
43

```

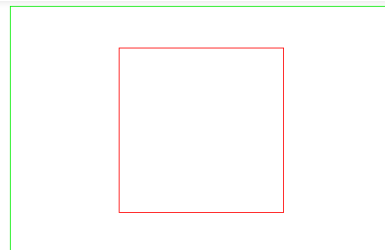


面试题：宽度一定的子类块状元素在父类下水平垂直居中

```

index.html
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <title>已知宽高实现盒子水平垂直居中</title>
7   <style type="text/css">
8     .box {
9       border: 1px solid #00ee00;
10      height: 300px;
11      position: relative;
12    }
13
14    .box1 {
15      width: 200px;
16      height: 200px;
17
18      border: 1px solid red;
19
20
21      position: absolute;
22
23      /* 这两句控制子元素左上角的位置 */
24      /* 配合绝对布局使用 */
25      /* 使得该点位于父元素的几何中心 */
26      top: 50%;
27      left: 50%;
28
29      /* 这一句控制 */
30      /* 配合盒子模型使用，margin是盒子模型最外层，-100px表示往上和左走100个像素 */
31      margin: -100px 0 0 -100px;
32    }
33  </style>
34 </head>
35
36 <body>
37   <div class="box">
38     <div class="box1"></div>
39   </div>
40 </body>
41
42 </html>
43

```



面试题：宽度不一定的子类块状元素在父类下水平垂直居中

- 1、利用父元素设置相对定位,子元素设置绝对定位,那么子元素就是相对于父元素定位的特性。
- 2、子元素设置上和左偏移的值都为50%。
- 3、然后再用css3属性translate位移,给上和左都位移-50%距离，就能达到垂直水平居中的效果。

