

一、浮动

[CSS基础总结（四） 浮动_css浮动-CSDN博客](#)

[CSS入门学习笔记+案例_定义1个h1标签,使用外部样式表,类选择器,维护h1标签的颜色和对齐方式-CSDN博客](#)

[网页制作基础学习——HTML+CSS常用代码_html网页制作代码大全-CSDN博客](#)

多个块级元素纵向排列用标准流，多个块级元素横向排列用浮动

所谓的标准流，也叫做文档流，就是标签按照规定好默认方式排列。

- 块级元素会独占一行，从上向下顺序排列：div、hr、p、h1~h6、ul、ol、dl、form、table 等
 - 行内元素会按照顺序，从左到右顺序排列，碰到父元素边缘则自动换行：span、a、i、em 等
- 以上都是标准流布局，标准流是最基本的布局方式。

但是标准流只能按照标签规定好的默认方式进行排列布局，面对特殊的场景，此时就需要“浮动”，因为浮动可以改变元素的默认排列方式，这使得网页布局更加灵活多变。

1.浮动的概念

float 属性用于创建浮动框，将其移动到一边，直到左边缘或右边缘触及包含块或另一个浮动框的边缘，这样即可使得元素进行“浮动”。

语法：

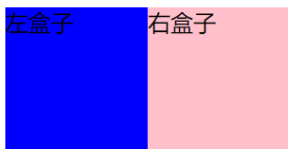
```
选择器 { float: 属性值; }
```

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

CSDN @AricL

浮动的典型应用： 让多个块级盒子（div）水平排列成一行

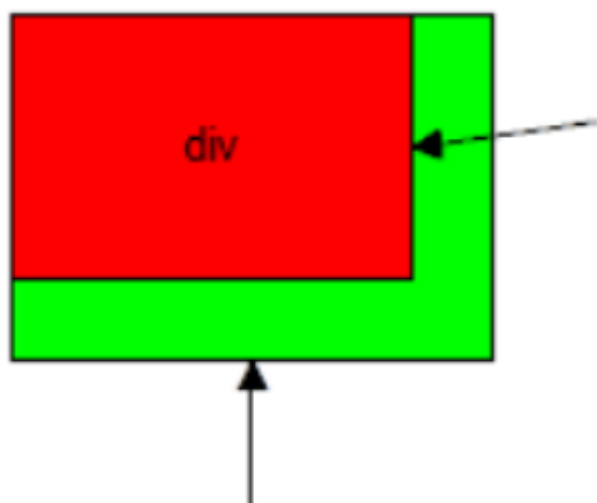
```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Document</title>
6      <style>
7          .left,.right {
8              /* 添加浮动，元素统一向左浮动，向右同理 float:right*/
9              float: left;
10             width: 100px;
11             height: 100px;
12             background-color: blue;
13         }
14         .right {
15             background-color: pink;
16         }
17     </style>
18 </head>
19 <body>
20     <div class="left">左盒子</div>
21     <div class="right">右盒子</div>
22 </body>
23 </html>
```



2.浮动的特性

2.1 脱标

所谓“脱标”，顾名思义就是脱离标准流。当元素设置了浮动后，该元素将会脱离标准流（普通流）的控制，移动到指定位置，浮动的盒子不再保留原来所占有的位置，就像飘浮在空中一样，脱离了原来的地面。



设置了浮动的元素，
漂浮在普通流的上面，
不占位置，脱标

正常标准流显示

CSDN @Aricl.

2.2 一行显示，顶部对齐

如果多个盒子都设置了浮动，则它们会按照属性值 一行内显示并且顶端对齐排列。

在一行显示，三个div都要设置浮动属性



CSDN @Aricl.

注意：

- 浮动的元素是互相贴靠在一起的（不会有缝隙），如果父级宽度装不下这些浮动的盒子，多出的盒子会另起一行对齐。
- 而如果使用元素模式转换，让块级元素转换成行内块元素，彼此之间是有空隙的。

2.3 具备行内块元素特性

不管原先是什么模式的元素，添加浮动之后具有行内块元素相似的特性

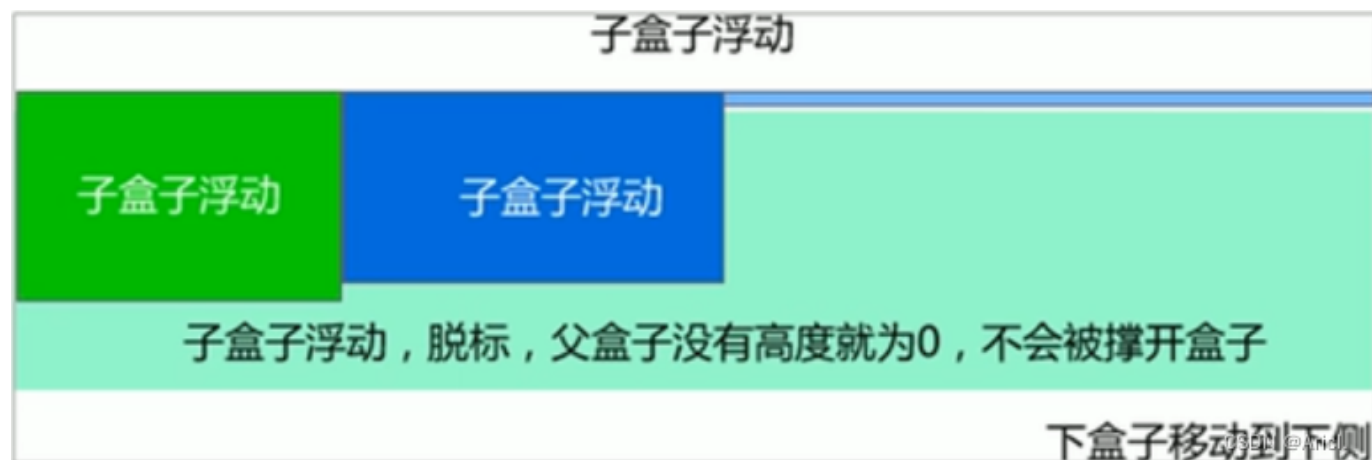
- 如果块级盒子没有设置宽度，默认宽度和父级一样宽，但是添加浮动后，它的大小根据内容来决定
- 浮动的盒子中间是没有缝隙的，是紧挨着的

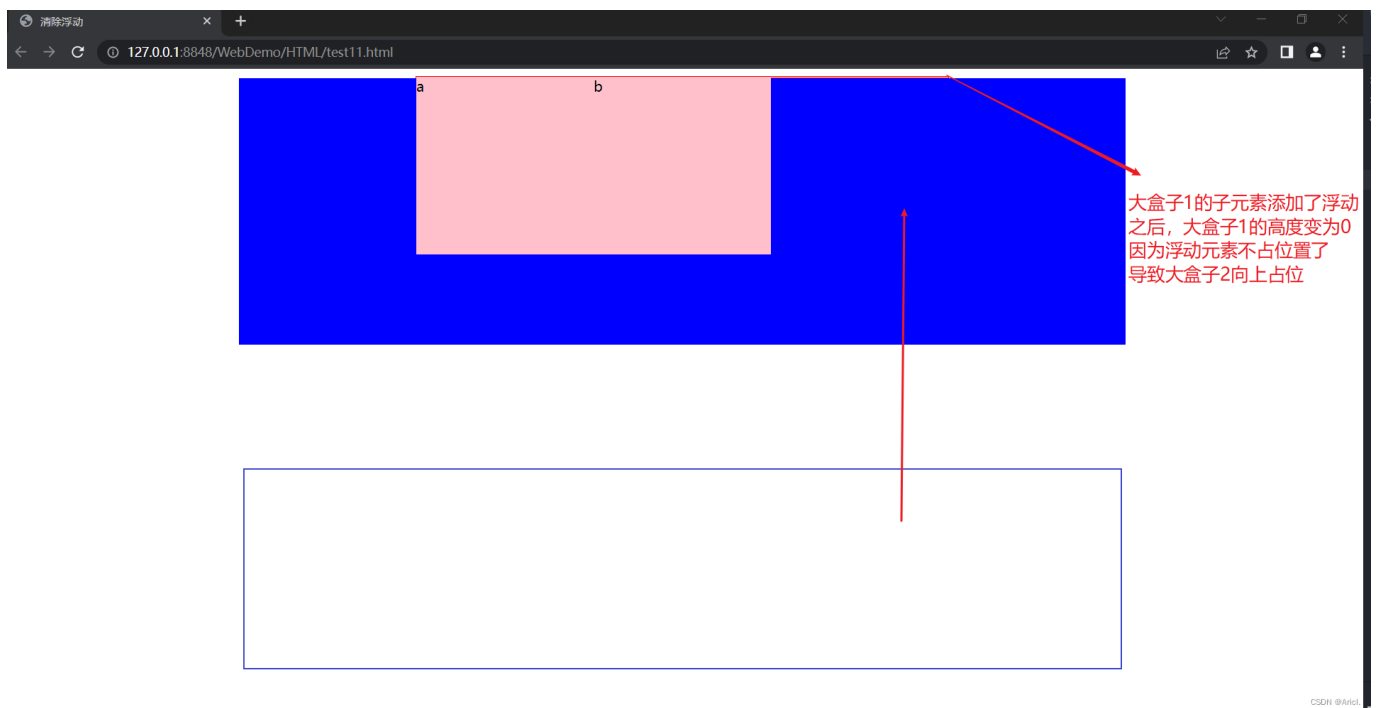
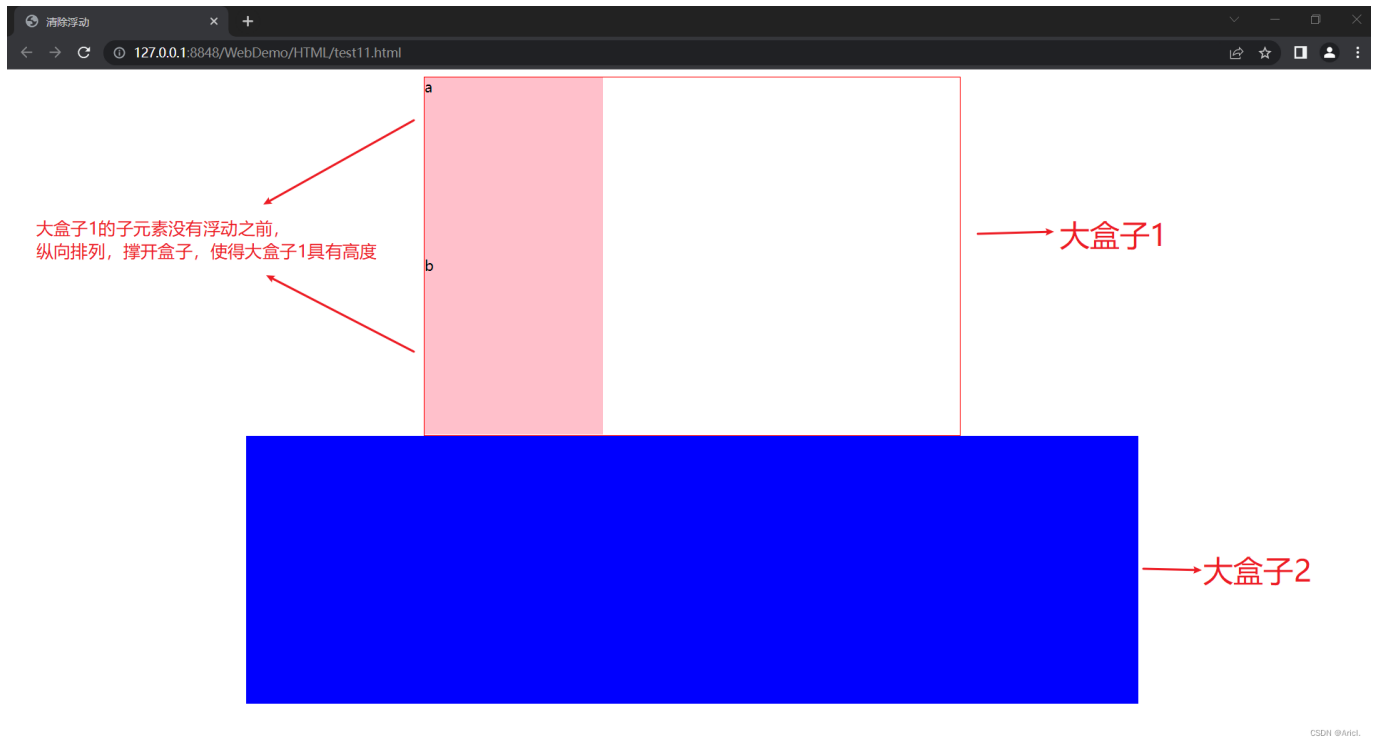
- 行内元素同理

3.清除浮动

```
选择器: {  
  clear: 属性值;  
}  
属性值:  
left: 在左侧不允许浮动元素  
right: 在右侧不允许浮动  
both: 在左右两侧均不允许浮动  
none: 默认值, 运行浮动出现在左右两侧
```

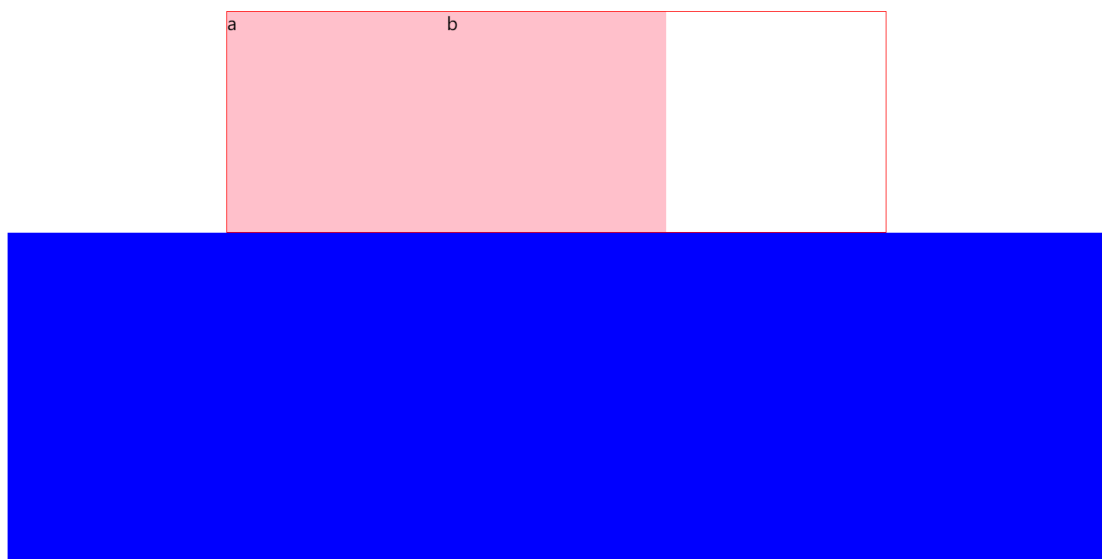
由于父级盒子很多情况下, 不方便给高度, 但是子盒子浮动又不占有位置, 最后父级盒子 高度为 0 时, 就会影响下面的标准流盒子, 从而对整个网页元素的排版产生不良影响。





清除浮动后：

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="utf-8">
5    <title>清除浮动</title>
6    <style>
7      .box1 {
8        width: 600px;
9        border: 1px solid red;
10       margin: 0 auto;
11     }
12     .son {
13       float: left;
14       width: 200px;
15       height: 200px;
16       background-color: pink;
17     }
18     .box2 {
19       width: 1000px;
20       height: 300px;
21       background-color: blue;
22       margin: 0 auto;
23     }
24     .clear1 {
25       clear: both;
26     }
27   </style>
28 </head>
29
30 <body>
31   <div class="box1">
32     <div class="son">a</div>
33     <div class="son">b</div>
34     <!-- 添加一个空盒子设置清除浮动的样式 -->
35     <div class="clear1"></div>
36   </div>
37   <div class="box2"></div>
38 </body>
</html>
```



4.清除浮动的方法

(1) 额外标签法（隔墙法）

隔墙法, 就是在 最后一个浮动的子元素 后面添加一个额外标签, 添加 清除浮动的样式 即可

语法:

```
<div style="clear:both"></div>
```

额外标签法的优缺点:

优点：通俗易懂，书写方便 缺点：添加了许多无意义的标签，结构化较差

(2) 父级添加overflow

可以给父级添加 overflow 属性，将其属性值设置为 hidden、 auto 或 scroll 。

```
选择器： {
  overflow: 属性值;
}
```

属性值：

visible:默认值，超出盒子自身的内容不会被修剪，会出现在元素框外

hidden:超出盒子自身的内容会被修剪，并且内容不可见

scroll:超出盒子自身的内容会被修剪，但是浏览器会显示滚动条以便查看其它内容

auto:如果超出盒子自身的内容会被修剪，则浏览器会显示滚动条以便查看其它内容

优缺点：

优点：代码非常简洁 缺点：无法显示溢出的部分

5.元素的显示和隐藏

5.1 display

通过display属性设置元素是否显示， 以及是否独占一行

常用取值：




取值	含义	说明
none	不显示	
inline	显示为内联元素，行级元素的默认值	将块级元素变为行级元素，不再独占一行
block	显示为块级元素，块级元素的默认值	将行级元素变为块级元素，独占一行
inline-block	显示为内联元素， 但是可以设置宽和高	在inline基础上允许设置宽度和高度

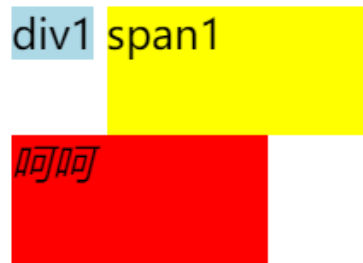
inline-block的优缺点如下： 1.可以让元素在一行， 支持宽高度， 代码实现起来既方便又便于理解， 并且属性的元素在标准文档流中， 无须清除浮动 2.位置方向不可控制， 会使排列在一行的元素中间有空格

浮动的优缺点如下： 1.可以让元素在一行， 支持高度和宽度， 可以决定排列方向 2.设置浮动后元素脱离标准文档流， 会对周围元素产生影响， 必须清除浮动

注意：

行级元素是无法设置宽度和高度的，可以为行级元素设置 display:inline-block ,然后就可以设置宽和高了

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Document</title>
6      <style>
7          .div1{
8              background:  lightblue;
9              display:inline;
10         }
11         span{
12             width: 100px;
13             height: 50px;
14             background:  yellow;
15             display:inline-block;
16         }
17         i{
18             display:block;
19             width: 100px;
20             height: 50px;
21             background:  red;
22         }
23     </style>
24 </head>
25 <body>
26     <div class="div1">div1</div>
27     <span>span1</span>
28     <i>呵呵</i>
29 </body>
30 </html>
```

5.2 visibility

也可以通过visibility属性设置元素的显示和隐藏

常用属性：

取值	含义	说明
visible	显示	
hidden	隐藏	

5.3 区别

- display隐藏时不再占据页面中的空间，后面的元素会占用其位置
- visibility隐藏时会占据页面中的空间，位置还保留在页面中，只是不显示

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Document</title>
6      <style>
7          p{
8              width: 50px;
9              height: 50px;
10             background:  red;
11         }
12         .p1{
13             visibility: hidden;
14         }
15         .p2{
16             display: none;
17         }
18     </style>
19 </head>
20 <body>
21     <p class="p1">p1hello</p>
22     <p >world1</p>
23     <p class="p2">p2hello</p>
24     <p >world2</p>
25 </body>
26 </html>
```

world1

world2

二、轮廓属性

1 简介

轮廓outline,用于在元素周围绘制一个轮廓，位于border外围，可以突出显示元素

2 基本用法

常用属性:

outline-width:轮廓宽度 outline-color:轮廓颜色 outline-style:轮廓样式 outline简写 在浏览器中，当鼠标单击或使用TAB键让一个表单或链接获得焦点时，该元素会有一个轮廓outline

优点：可以提高使用表单的用户体验

缺点：有时会影响美观

3 outline和border的区别

border可以应用于所有html元素，而outline主要用于表单元素、超链接元素 当元素获得焦点时会自动出现outline轮廓效果，当失去焦点时会自动消失，这是浏览器默认行为 outline不影响元素的尺寸和位置，而border会影响

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Document</title>
6      <style>
7          span{
8              border:2px solid red;
9              outline:4px dashed green;
10         }
11         .username{
12             border:1px solid red;
13             outline:none;
14             padding-left:3px;
15             width:80px;
16         }
17         .email{
18             border:0;
19             outline:0;
20             border-bottom:1px solid #000;
21         }
22     </style>
23 </head>
24 <body>
25     <span>welcome to CSS</span>
26     <hr>
27     用户名: <input type="text" class="username">
28     <hr>
29     邮箱: <input type="text" class="email">
30 </body>
31 </html>

```

welcome to CSS

用户名:

邮箱:

三、cursor属性

用于设置光标的形状

常用属性：

- default默认光标，一般为箭头
- pointer手形，光标移动超链接上时一般显示为手形
- move表示可移动
- text表示文 本
- wait表示程序正忙，需要等待
- help表示帮助

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Document</title>
6  <style>
7      span{
8          cursor:help;
9      }
10     p{
11         cursor: pointer;
12     }
13 </style>
14 </head>
15 <body>
16     <span>光标形状-help</span>
17     <p>光标形状-pointer</p>
18 </body>
19 </html>
```

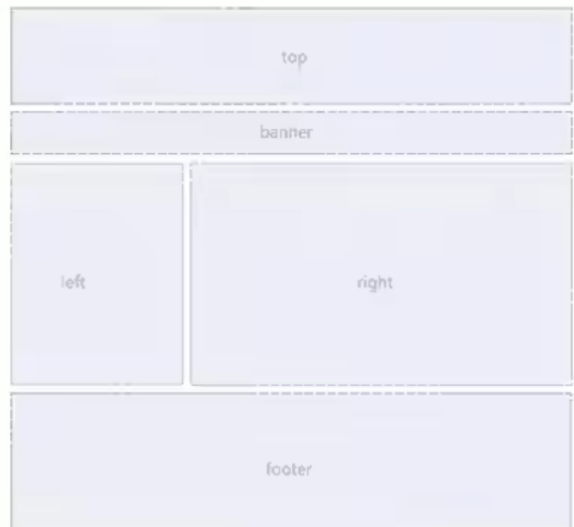
四、常见网页布局案例

1.基本的网页布局

盒子从上到下排列布局，或者中间主体部分分为左右两部分等等



基本网页布局1



基本网页布局2

CSDN @Ariel.

基本网页布局1:

```

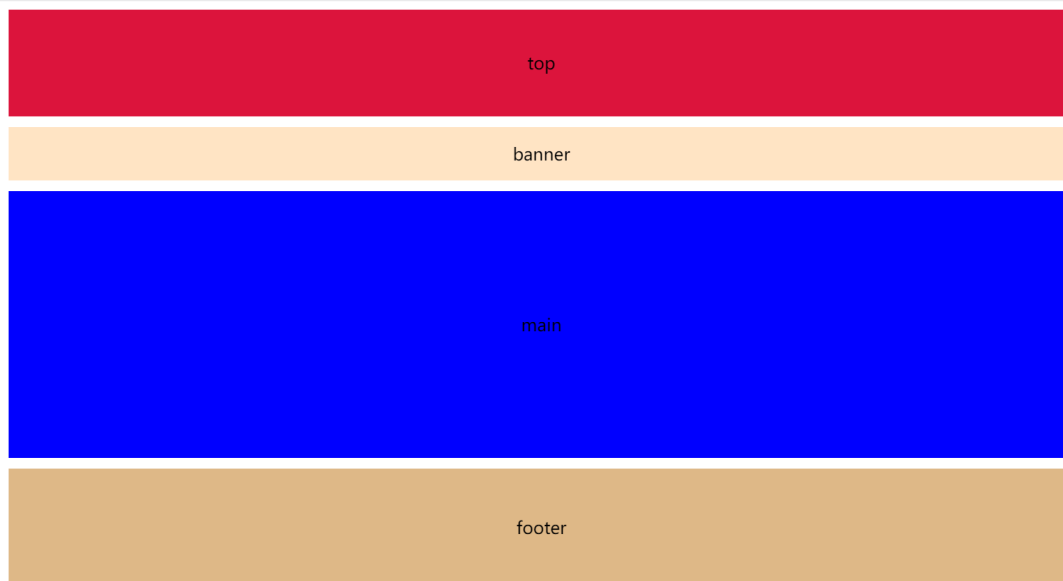
6      <style>
7      .top {
8          height: 100px;
9          background-color: crimson;
10         margin: 0 auto;
11         text-align: center;
12         line-height: 100px;
13     }
14     .banner {
15         height: 50px;
16         background-color: bisque;
17         margin: 10px auto;
18         text-align: center;
19         line-height: 50px;
20     }
21     .main {
22         height: 250px;
23         background-color: blue;
24         margin: 10px auto;
25         text-align: center;
26         line-height: 250px;
27     }
28     .footer {
29         height: 110px;
30         background-color: burlywood;
31         margin: 0 auto;
32         text-align: center;
33         line-height: 110px;
34     }
35     div {
36         width: 1000px;
37     }

```

```

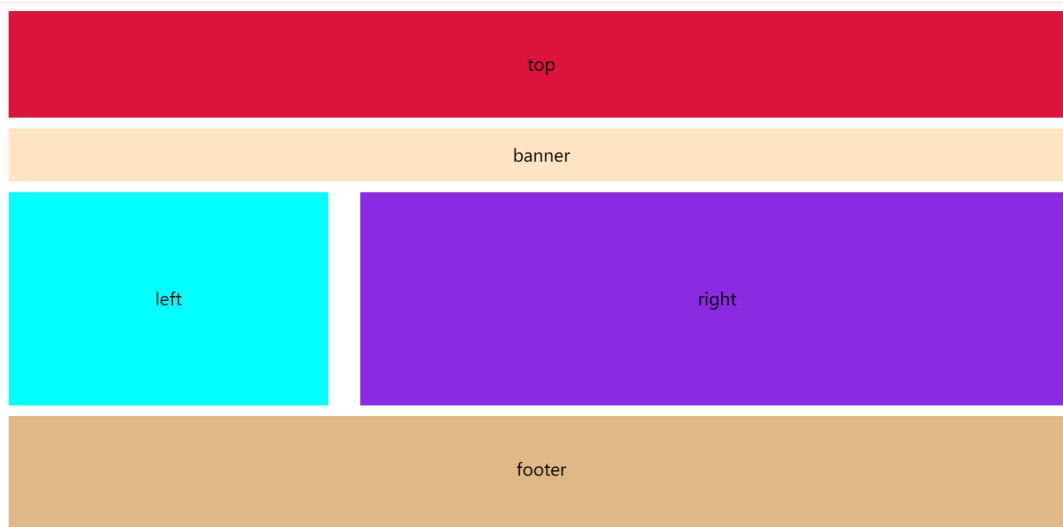
38     </style>
39 </head>
40 <body>
41     <div class="top">top</div>
42     <div class="banner">banner</div>
43     <div class="main">main</div>
44     <div class="footer">footer</div>
45 </body>
46 </html>

```



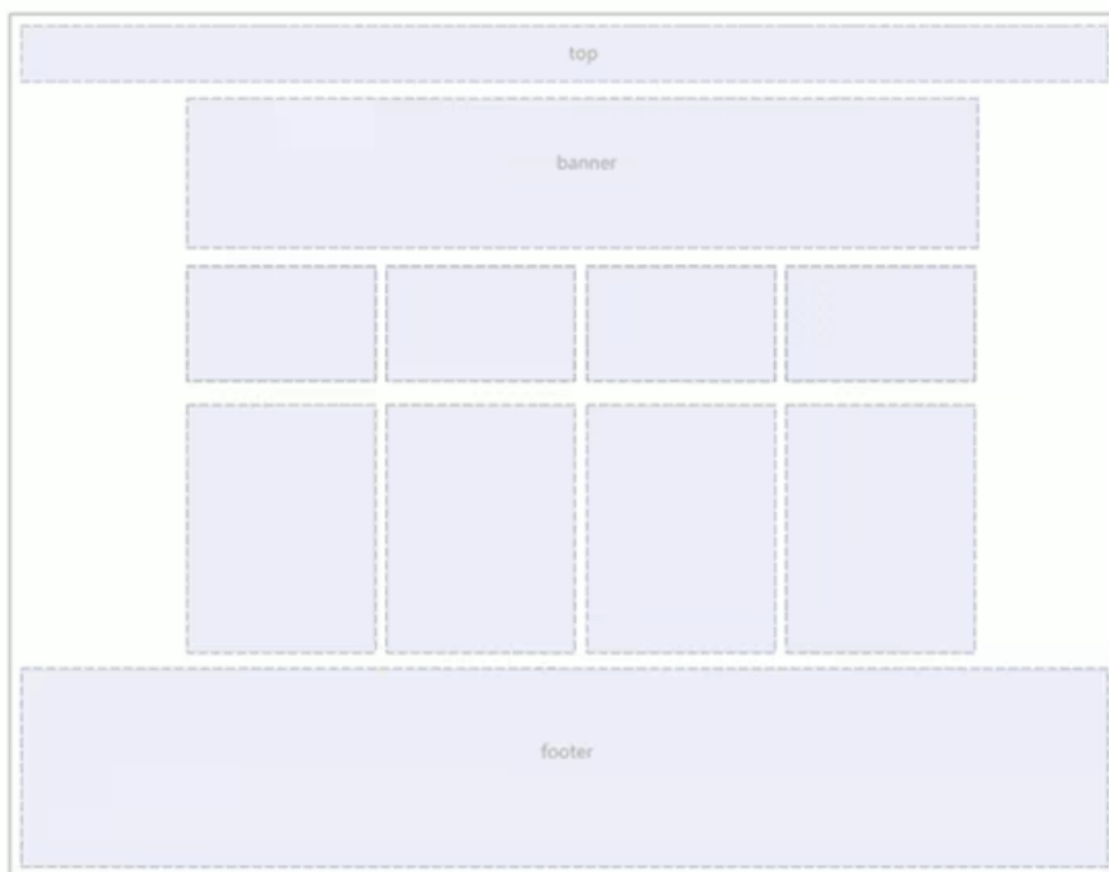
基本网页布局2:

```
5 <title>基本网页布局2</title>
6 <style>
7     .top {
8         width: 1000px;
9         height: 100px;
10        background-color: crimson;
11        margin: 0 auto;
12        text-align: center;
13        line-height: 100px;
14    }
15    .banner {
16        width: 1000px;
17        height: 50px;
18        background-color: bisque;
19        margin: 10px auto;
20        text-align: center;
21        line-height: 50px;
22    }
23    .main {
24        width: 1000px;
25        overflow: hidden;
26        margin: 10px auto;
27    }
28    .footer {
29        width: 1000px;
30        height: 110px;
31        background-color: burlywood;
32        margin: 0 auto;
33        text-align: center;
34        line-height: 100px;
35    }
36
37    .son1 {
38        float: left;
39        width: 300px;
40        height: 200px;
41        background-color: aqua;
42        margin-right: 30px;
43        text-align: center;
44        line-height: 200px;
45    }
46    .son2 {
47        float: left;
48        width: 670px;
49        height: 200px;
50        background-color: blueviolet;
51        text-align: center;
52        line-height: 200px;
53    }
54 </style>
55 <body>
56     <div class="top">top</div>
57     <div class="banner">banner</div>
58     <div class="main">
59         <div class="son1">left</div>
60         <div class="son2">right</div>
61     </div>
62     <div class="footer">footer</div>
63 </body>
64 </html>
```



2.更细致的网页布局

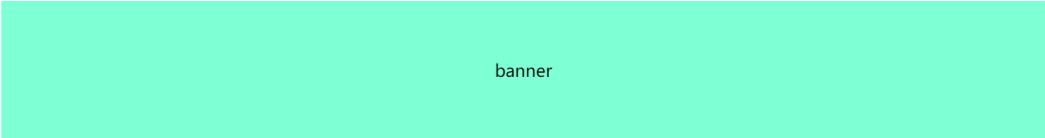
更为细致一点的网页布局，主要体现在中间主体部分模块数增多，顶部和底部变为通栏，即和浏览器的长度一致。



更加细致的网页布局


```
7  .top {
8      width: 100%;
9      height: 50px;
10     background-color: crimson;
11     text-align: center;
12     line-height: 50px;
13 }
14 .banner {
15     width: 980px;
16     height: 130px;
17     background-color: aquamarine;
18     line-height: 130px;
19     text-align: center;
20     margin: 10px auto;
21 }
22 .main1 {
23     width: 980px;
24     margin: 10px auto;
25     overflow: hidden;
26 }
27 .main1 div {
28     float: left;
29     width: 237px;
30     height: 100px;
31     background-color: blue;
32     margin-right: 10px;
33     text-align: center;
34     line-height: 100px;
35 }
36 .main1 .last {
37     margin: 0 ;
38 }
39
40 .main2 {
41     width: 980px;
42     margin: 10px auto;
43     overflow: hidden;
44 }
45 .main2 div {
46     float: left;
47     width: 237px;
48     height: 170px;
49     background-color: blue;
50     margin-right: 10px;
51     text-align: center;
52     line-height: 170px;
53 }
54 .main2 .last {
55     margin: 0 ;
56 }
57
58 .footer {
59     width: 100%;
60     height: 100px;
61     background-color: crimson;
62     text-align: center;
63     line-height: 100px;
64 }
65
66 </style>
67 </head>
68 <body>
69     <div class="top">top</div>
70     <div class="banner">banner</div>
71     <div class="main1">
72         <div>a</div>
73         <div>b</div>
74         <div>c</div>
75         <div class="last">d</div>
76     </div>
77     <div class="main2">
78         <div>e</div>
79         <div>f</div>
80         <div>g</div>
81         <div class="last">h</div>
82     </div>
83     <div class="footer">footer</div>
84 </body>
85 </html>
```

```
<body>
  <div class="top">top</div>
  <div class="banner">banner</div>
  <div class="main1">
    <div>a</div>
    <div>b</div>
    <div>c</div>
    <div class="last">d</div>
  </div>
  <div class="main2">
    <div>e</div>
    <div>f</div>
    <div>g</div>
    <div class="last">h</div>
  </div>
  <div class="footer">footer</div>
</body>
</html>
```



a	b	c	d
e	f	g	h