

# 显示与隐藏

CSS 中的 display 和 visibility 都可以设置一个元素在浏览器中的显示或隐藏效果。

- display: 隐藏某个元素时，不会占用任何空间。换句话说讲，不会影响布局。
- visibility: 隐藏某个元素时，仍需占用与未隐藏之前一样的空间。换句话说讲，会影响布局。

## display 属性

属性值	描述
none	此元素不会被显示。

当将一个元素的 CSS 属性 display 设置为 none 时，该元素会被隐藏。并且被隐藏的元素不会占用 HTML 页面的任何空间。

在 HTML 页面中定义两个 `<div>` 标签，并设置 CSS 样式，具体代码如下：

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title></title>
6      <style>
7        #d1 {
8          width: 300px;
9          height: 300px;
10         background-color: lightskyblue;
11       }
12       #d2 {
13         width: 300px;
14         height: 300px;
15         background-color: green;
16       }
17     </style>
18   </head>
19   <body>
20     <div id="d1"></div>
21     <div id="d2"></div>
22   </body>
23 </html>
```

上述 HTML 页面运行的效果如下：



当为第一个 `<div>` 标签设置 `display` 属性值为 `none` 时，该 `<div>` 标签会被隐藏。



属性值	描述
block	此元素将显示为块级元素。

该值主要作用于**内联元素**。如果将内联元素的 CSS 属性 `display` 设置为 `block` 时，该内联元素将在浏览器中显示成块级元素效果。

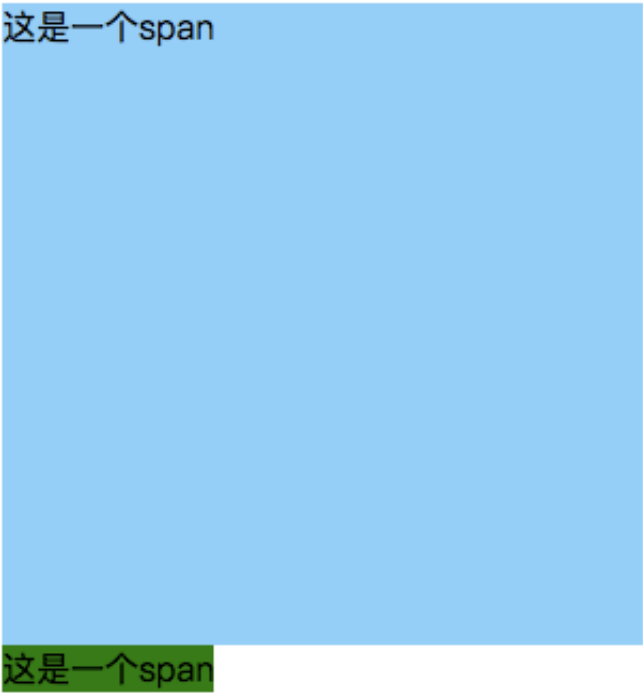
在 HTML 页面中定义两个 `<div>` 标签，并设置 CSS 样式，具体代码如下：

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6     <style>
7       #s1 {
8         width: 300px;
9         height: 300px;
10        background-color: lightskyblue;
11      }
12      #s2 {
13        width: 300px;
14        height: 300px;
15        background-color: green;
16      }
17    </style>
18  </head>
19  <body>
20    <span id="s1">这是一个span</span>
21    <span id="s2">这是一个span</span>
22  </body>
23 </html>
```

上述 HTML 页面运行的效果如下：

这是一个span 这是一个span

当为第一个 `<span>` 标签设置 `display` 属性值为 `block` 时，该 `<span>` 标签会呈现块级元素效果。



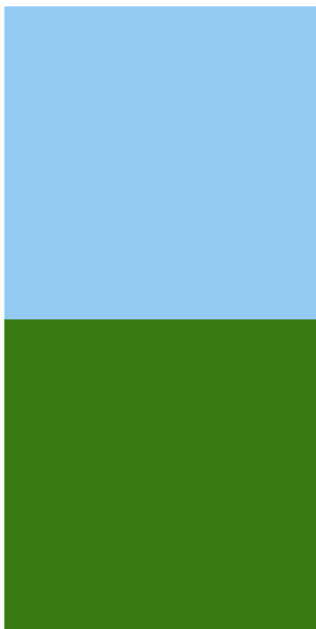
属性值	描述
inline	默认值，此元素会被显示为内联元素。

该值主要作用于**块级元素**。如果将内联元素的 CSS 属性 display 设置为 inline 时，该内联元素将在浏览器中显示成内联元素效果。

在 HTML 页面中定义两个 `<div>` 标签，并设置 CSS 样式，具体代码如下：

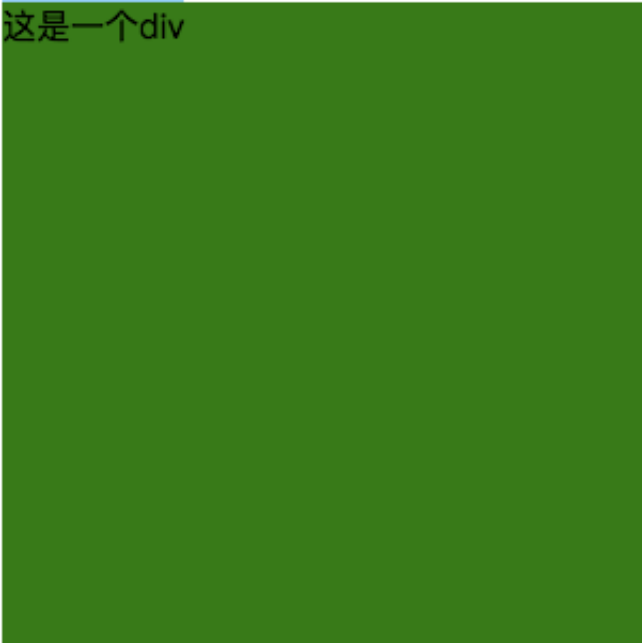
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6     <style>
7       #d1 {
8         width: 300px;
9         height: 300px;
10        background-color: lightskyblue;
11      }
12      #d2 {
13        width: 300px;
14        height: 300px;
15        background-color: green;
16      }
17    </style>
18  </head>
19  <body>
20    <div id="d1">这是一个div</div>
21    <div id="d2">这是一个div</div>
22  </body>
23 </html>
```

上述 HTML 页面运行的效果如下:



当为第一个 `<div>` 标签设置 `display` 属性值为 `inline` 时, 该 `<div>` 标签会呈现内联元素样式。

这是一个div  
这是一个div



---

属性值	描述
inline-block	行内块元素。

当将 CSS 属性 display 设置为 inline-block 时，该元素在浏览器中显示为内联块级效果。即每个元素呈现块级元素效果，元素之间呈现内联元素效果。

在 HTML 页面中定义两个 `<div>` 标签，并设置 CSS 样式，具体代码如下：

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6     <style>
7       #d1 {
8         width: 300px;
9         height: 300px;
10        background-color: lightskyblue;
11      }
12      #d2 {
13        width: 300px;
14        height: 300px;
15        background-color: green;
16      }
17    </style>
18  </head>
19  <body>
20    <div id="d1">这是一个div</div>
21    <div id="d2">这是一个div</div>
22  </body>
23 </html>
```

上述 HTML 页面运行的效果如下：



分别为两个 `<div>` 标签设置 `display` 属性值为 `inline-block` 时，这两个 `<div>` 会显示在一行中。

这是一个div

这是一个div

## visibility 属性

visibility 属性指定一个元素是否是可见的。

值得注意的是: visibility 属性设置元素不可见的元素，但会占据页面上的空间。请使用 display 属性来创建不占据页面空间的不可见元素。

属性值	描述
visible	默认值，元素是可见的。
hidden	元素是不可见的。

在 HTML 页面中定义两个 `<div>` 标签，并设置 CSS 样式，具体代码如下：



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6     <style>
7       #d1 {
8         width: 300px;
9         height: 300px;
10        background-color: lightskyblue;
11      }
12      #d2 {
13        width: 300px;
14        height: 300px;
15        background-color: green;
16      }
17    </style>
18  </head>
19  <body>
20    <div id="d1">这是一个div</div>
21    <div id="d2">这是一个div</div>
22  </body>
23 </html>
```

上述 HTML 页面运行的效果如下:



当为第一个 `<div>` 标签设置 `visibility` 属性值为 `hidden` 时, 该 `<div>` 会被隐藏。

这是一个div



## overflow 属性

overflow 属性用于设置当元素中的内容溢出后的情况。

**值得注意的是:** 所谓溢出，是指子元素的大小（包括文本、元素或图片等）超出父元素的区域，会有一些内容显示在父元素所在的区域外。

属性值	描述
visible	默认值。内容不会被修剪，会呈现在元素框之外。
hidden	内容会被修剪，并且其余内容是不可见的。

上述两个值，要么设置溢出的内容默认显示在父元素的区域之外，要么设置溢出的内容隐藏不可见。这两种结果都是很好的效果。所以，在实际开发中，很少使用这两个值。

在 HTML 页面中定义一个 `<div>` 标签，设置 CSS 样式，并设置较多的文本内容，具体代码如下：

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6     <style>
7       div {
8         width: 300px;
9         height: 300px;
10        background-color: lightskyblue;
11      }
12    </style>
13  </head>
14  <body>
15    <div>
16      在我的后园，可以看见墙外有两株树，一株是枣树，还有一株也是枣树。
17      这上面的夜的天空，奇怪而高，我生平没有见过这样奇怪而高的天空。他仿佛要离开人间而去，使人们仰面
18      我不知道那些花草真叫什么名字，人们叫他们什么名字。我记得有一种开过极细小的粉红花，现在还开着，
19      枣树，他们简直落尽了叶子。先前，还有一两个孩子来打他们，别人打剩的枣子，现在是一个也不剩了，连
20      鬼映眼的天空越加非常之蓝，不安了，仿佛想离去人间，避开枣树，只将月亮剩下。然而月亮也暗暗地躲到
21      哇的一声，夜游的恶鸟飞过了。
22      我忽而听到夜半的笑声，吃吃地，似乎不愿意惊动睡着的人，然而四围的空气都应和着笑。夜半，没有别的
23      后窗的玻璃上丁丁地响，还有许多小飞虫乱撞。不多久，几个进来了，许是从窗纸的破孔进来的。他们一进来
24      猩红的栀子开花时，枣树又要做小粉红花的梦，青葱地弯成弧形了.....我又听到夜半的笑声；我赶紧砍断我的
25      我打一个呵欠，点起一支纸烟，喷出烟来，对着灯默默地敬奠这些苍翠精致的英雄们。
26      一九二四年九月十五日。
27    </div>
28  </body>
29 </html>
```

上述 HTML 页面运行的效果如下：

在我的后园，可以看见墙外有两株树，一株是枣树，还有一株也是枣树。这上面的夜的天空，奇怪而高，我生平没有见过这样奇怪而高的天空。他仿佛要离开人间而去，使人们仰面不再看见。然而现在却非常之蓝，闪闪地映着几十个星星的眼，冷眼。他的口角上现出微笑，似乎自以为大有深意，而将繁霜洒在我的园里的野花草上。我不知道那些花草真叫什么名字，人们叫他们什么名字。我记得有一种开过极细小的粉红花，现在还开着，但是更极细小了，她在冷的夜气中，瑟缩地做梦，梦见春的到来，梦见秋的到来，梦见瘦的诗人将眼泪擦在她最末的花瓣上，告诉她秋虽然来，冬虽然来，而此后接着还是春，蝴蝶乱飞，蜜蜂都唱起春词来了。她于是——笑，虽然颜色冻得红惨惨地，仍然瑟缩着。枣树，他们简直落尽了叶子。先前，还有一两个孩子来打他们，别人打剩的枣子，现在是一个也不剩了，连叶子也落尽了。他知道小粉红花的梦，秋后要有春；他也知道落叶的梦，春后还是秋。他简直落尽叶子，单剩干子，然而脱了当初满树是果实和叶子时候的弧形，欠伸得很舒服。但是，有几枝还低亚着，护定他从打枣的竿梢所得的皮伤，而最直最长的几枝，却已默默地铁似的直刺着奇怪而高的天空，使天空闪闪地鬼眨眼；直刺着天空中圆满的月亮，使月亮窘得发白。鬼眨眼的天空越加非常之蓝，不安了，仿佛想离

当为第一个 `<div>` 标签设置 `overflow` 属性值为 `hidden` 时，该 `<div>` 中的溢出文本内容会被隐藏。

在我的后园，可以看见墙外有两株树，一株是枣树，还有一株也是枣树。这上面的夜的天空，奇怪而高，我生平没有见过这样奇怪而高的天空。他仿佛要离开人间而去，使人们仰面不再看见。然而现在却非常之蓝，闪闪地映着几十个星星的眼，冷眼。他的口角上现出微笑，似乎自以为大有深意，而将繁霜洒在我的园里的野花草上。我不知道那些花草真叫什么名字，人们叫他们什么名字。我记得有一种开过极细小的粉红花，现在还开着，但是更极细小了，她在冷的夜气中，瑟缩地做梦，梦见春的到来，梦见秋的到来，梦见瘦的诗人将眼泪擦在她最末的花瓣上，告诉她秋

属性值	描述
scroll	内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。
auto	如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。

在 HTML 页面中定义一个 `<div>` 标签，设置 CSS 样式，并设置较多的文本内容，具体代码如下：

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6     <style>
7       div {
8         width: 300px;
9         height: 300px;
10        background-color: lightskyblue;
11      }
12    </style>
13  </head>
14  <body>
15    <div>
16      在我的后园，可以看见墙外有两株树，一株是枣树，还有一株也是枣树。
17      这上面的夜的天空，奇怪而高，我生平没有见过这样奇怪而高的天空。他仿佛要离开人间而去，使人们仰面
18      我不知道那些花草真叫什么名字，人们叫他们什么名字。我记得有一种开过极细小的粉红花，现在还开着，
19      枣树，他们简直落尽了叶子。先前，还有一两个孩子来打他们，别人打剩的枣子，现在是一个也不剩了，连
20      鬼映眼的天空越加非常之蓝，不安了，仿佛想离去人间，避开枣树，只将月亮剩下。然而月亮也暗暗地躲到
21      哇的一声，夜游的恶鸟飞过了。
22      我忽而听到夜半的笑声，吃吃地，似乎不愿意惊动睡着的人，然而四围的空气都应和着笑。夜半，没有别的
23      后窗的玻璃上丁丁地响，还有许多小飞虫乱撞。不多久，几个进来了，许是从窗纸的破孔进来的。他们一进来
24      猩红的栀子开花时，枣树又要做小粉红花的梦，青葱地弯成弧形了.....我又听到夜半的笑声；我赶紧砍断我的
25      我打一个呵欠，点起一支纸烟，喷出烟来，对着灯默默地敬奠这些苍翠精致的英雄们。
26      一九二四年九月十五日。
27    </div>
28  </body>
29 </html>
```

上述 HTML 页面运行的效果如下：

在我的后园，可以看见墙外有两株树，一株是枣树，还有一株也是枣树。这上面的夜的天空，奇怪而高，我生平没有见过这样奇怪而高的天空。他仿佛要离开人间而去，使人们仰面不再看见。然而现在却非常之蓝，闪闪地映着几十个星星的眼，冷眼。他的口角上现出微笑，似乎自以为大有深意，而将繁霜洒在我的园里的野花草上。我不知道那些花草真叫什么名字，人们叫他们什么名字。我记得有一种开过极细小的粉红花，现在还开着，但是更极细小了，她在冷的夜气中，瑟缩地做梦，梦见春的到来，梦见秋的到来，梦见瘦的诗人将眼泪擦在她最末的花瓣上，告诉她秋虽然来，冬虽然来，而此后接着还是春，蝴蝶乱飞，蜜蜂都唱起春词来了。她于是是一笑，虽然颜色冻得红惨惨地，仍然瑟缩着。枣树，他们简直落尽了叶子。先前，还有一两个孩子来打他们，别人打剩的枣子，现在是一个也不剩了，连叶子也落尽了。他知道小粉红花的梦，秋后要有春；他也知道落叶的梦，春后还是秋。他简直落尽叶子，单剩干子，然而脱了当初满树是果实和叶子时候的弧形，欠伸得很舒服。但是，有几枝还低亚着，护定他从打枣的竿梢所得的皮伤，而最直最长的几枝，却已默默地铁似的直刺着奇怪而高的天空，使天空闪闪地鬼眨眼；直刺着天空中圆满的月亮，使月亮窘得发白。鬼眨眼的天空越加非常之蓝，不安了，仿佛想离

当为第一个 `<div>` 标签设置 `overflow` 属性值为 `scroll` 时，该 `<div>` 中的溢出文本内容会显示滚动条。

蝴蝶乱飞，蜜蜂都唱起春词来了。她于是是一笑，虽然颜色冻得红惨惨地，仍然瑟缩着。枣树，他们简直落尽了叶子。先前，还有一两个孩子来打他们，别人打剩的枣子，现在是一个也不剩了，连叶子也落尽了。他知道小粉红花的梦，秋后要有春；他也知道落叶的梦，春后还是秋。他简直落尽叶子，单剩干子，然而脱了当初满树是果实和叶子时候的弧形，欠伸得很舒服。但是，有几枝还低亚着，护定他从打枣的竿梢所得的皮伤，而最直最长的几枝，却已默默地铁似的直刺着奇怪而高的天空，使天空闪闪地鬼眨眼；直刺着天空中圆满的月亮，使月亮窘得发白。鬼眨眼

上述两个值，当有溢出的内容出现时，都会显示滚动条以便查看其余的内容。但不同是：

- `scroll`: 无论是否有溢出的内容，始终显示滚动条。
- `auto`: 当有溢出的内容出现时，才显示滚动条。如果没有溢出的内容时，会自动隐藏滚动条。

## 文档流

将窗体自上而下分成一行一行，并在每行中按从左至右的挨次排放元素，即为文档流。

文档流是 HTML 页面的底层结构，HTML 页面创建的元素默认都在文档流中。

- 块级元素

- 块级元素在文档流中自上向下排列（垂直方向排列）
- 块级元素在文档流中默认的宽度是父元素的100%
- 块级元素在文档流中默认的高度是所有子元素的高度总和
- 内联元素
  - 内联元素在文档流中自左向右水平排列（水平方向排列）
  - 内联元素的宽度和高度都由内容确定

**值得注意的是:** 如果在一行中不能容纳所有的元素，则会换到下一行，继续自左向右排列。

有三种状况将使得元素离开文档流而存在，分别是浮动、绝对定位、固定定位。

**值得注意的是:** 在 IE 浏览器中浮动元素也存在于文档流中。

## 浮动

---

CSS 的 Float（浮动），会使元素脱离文档流，向左或向右移动。其他元素会被重新排列。

Float（浮动），往往是用于图像，但它在布局时一样非常有用。

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

设置元素浮动效果时，以下四种情况比较特殊：

- 子元素只能在父元素的区域中进行浮动（子元素浮动不能超出父元素的范围）。

在 HTML 页面中定义两个 `<div>` 标签为父子元素，并设置 CSS 样式，具体代码如下：

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6     <style>
7       #d1 {
8         height: 300px;
9         background-color: lightskyblue;
10      }
11      #d2 {
12        width: 50px;
13        height: 50px;
14        background-color: green;
15      }
16    </style>
17  </head>
18  <body>
19    <div id="d1">
20      <div id="d2"></div>
21    </div>
22  </body>
23 </html>
```

上述 HTML 页面运行的效果如下:



当为子元素 `<div>` 标签设置 `float` 属性值为 `right` 时, 该 `<div>` 为浮动效果。





- 兄弟元素同时设置相同浮动效果时，会进行自动排列。

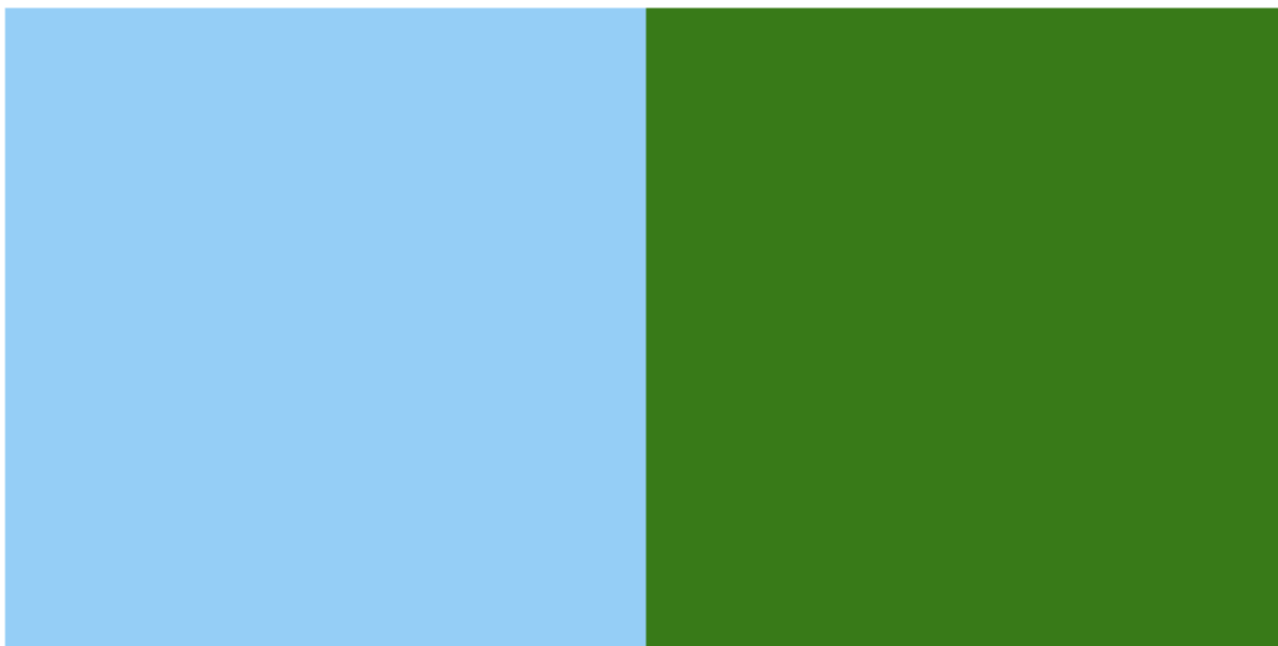
在 HTML 页面中定义两个 `<div>` 标签为兄弟元素，并设置 CSS 样式，具体代码如下：

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6     <style>
7       #d1 {
8         width: 300px;
9         height: 300px;
10        background-color: lightskyblue;
11      }
12      #d2 {
13        width: 300px;
14        height: 300px;
15        background-color: green;
16      }
17    </style>
18  </head>
19  <body>
20    <div id="d1"></div>
21    <div id="d2"></div>
22  </body>
23 </html>
```

上述 HTML 页面运行的效果如下：



当同时为两个 `<div>` 标签设置 float 属性值为 left 时，这两个 `<div>` 为浮动效果。



- 如果兄弟元素中，前一个兄弟元素没有设置浮动，而后一个兄弟元素设置浮动的话，后一个兄弟元素的浮动不能超过前一个兄弟元素。

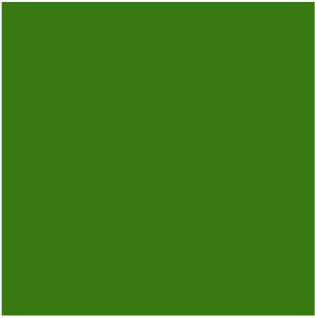
在 HTML 页面中定义两个 `<div>` 标签为兄弟元素，并设置 CSS 样式，具体代码如下：

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6     <style>
7       #d1 {
8         width: 300px;
9         height: 300px;
10        background-color: lightskyblue;
11      }
12      #d2 {
13        width: 300px;
14        height: 300px;
15        background-color: green;
16      }
17    </style>
18  </head>
19  <body>
20    <div id="d1"></div>
21    <div id="d2"></div>
22  </body>
23 </html>
```

上述 HTML 页面运行的效果如下:



当为第二个 `<div>` 标签设置 `float` 属性值为 `right` 时, 该 `<div>` 为浮动效果。



**值得注意的是：**所有浮动元素的层级要高于文档流中的元素的层级。但文字是个例外，文字不会被浮动元素所覆盖，而是环绕在浮动元素的周围。

## 块级元素浮动

块级元素设置浮动之后，会从文档流中脱离，并且具有以下特点：

- 块级元素不会独占 HTML 页面的一行。换句话讲，多个块级元素可以处在一行。
- 块级元素的宽度和高度等于所有后代元素的总和。

## 内联元素浮动

内联元素设置浮动之后，也会从文档流中脱离，并且会呈现块级元素的效果。特点也与设置浮动的块级元素相同。

## 清除浮动

clear 属性指定元素的左侧或右侧不允许浮动的元素。

属性值	描述
none	默认值。允许浮动元素出现在两侧。
left	在左侧不允许浮动元素。
right	在右侧不允许浮动元素。
both	在左右两侧均不允许浮动元素。

# 高度塌陷

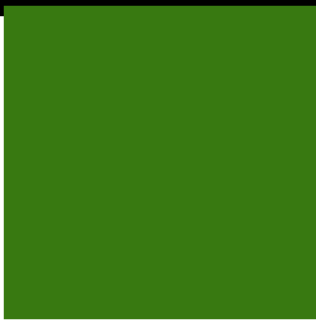
在 HTML 页面中定义两个 `<div>` 标签为父子元素，并设置 CSS 样式，具体代码如下：

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6     <style>
7       #d1 {
8         border: 10px black solid;
9         background-color: lightskyblue;
10      }
11      #d2 {
12        width: 300px;
13        height: 300px;
14        background-color: green;
15      }
16    </style>
17  </head>
18  <body>
19    <div id="d1">
20      <div id="d2"></div>
21    </div>
22  </body>
23 </html>
```

上述 HTML 页面运行的效果如下：



为作为子元素 `<div>` 设置 float 属性值为 left 时，该 `<div>` 为浮动效果。



由于并没有为作为父元素的 `<div>` 设置高度，作为父元素的 `<div>` 实际的高度与作为子元素的 `<div>` 的高度相等。

但作为子元素的 `<div>` 设置浮动后，脱离了文档流，导致作为父元素的 `<div>` 没有了高度。

现在所看到的效果，就被称为**高度塌陷**。

所谓**高度塌陷**，就是指在没有为父元素设置指定高度，并且设置子元素为浮动时，父元素的高度丢失的情况。

高度塌陷的问题，可以通过 BFC（Block Formatting Context）块级格式化环境进行解决。

## BFC 块级格式化环境

BFC（Block Formatting Context）是元素的隐含属性。默认情况下 BFC 是关闭的，当元素开启 BFC 以后，将会具有如下特性：

- 文档流中的元素不会被设置为浮动的元素所覆盖。
- 子元素的垂直方向的外边距不会传递给父元素。
- 元素可以包含浮动的子元素。

当然，并不能直接去开启 BFC，而需要使用一些特殊的样式来间接的打开元素的 BFC。

- 设置元素为浮动（float）
- 设置元素的 display 为 inline-block
- 设置元素为绝对定位（后面的内容）
- 将元素的 overflow 设置为一个非 visible 的值（一般设置为 hidden）
- 在所有子元素的最后新增一个子元素，并设置 clear 属性值为 both

值得注意的是：开启 BFC 都会有一些副作用，需要选择一些副作用小的方式。

---

## 版权说明

本笔记的内容免费开源，任何人都可以免费学习、分享，甚至可以进行修改。但需要注明作者及来源，并且不能用于商业。

本笔记采用[知识共享署名-非商业性使用-禁止演绎 4.0 国际许可协议](#)进行许可。

# 联系方式

---



可以问职业问题。  
不要问技术问题。



可以问职业问题。  
可以问技术问题。



可以问职业问题。  
可以问技术问题。

