我们先提出一个问题：在页面上有两个盒子，一个盒子靠着屏幕的左侧显示，另一个盒子靠着屏幕的右侧显示，如果让你们实现这个效果，可不可以呢？

好，我们来尝试一下。新建一个“01-两个盒子一左一右显示在页面的同一行”，大家现在有没有什么好主意呢？

我们先来画一张图：这是一个浏览器窗口，这是一个盒子，这是另外一个盒子，默认情况下，它们是不是一上一下？我现在要的效果是一个靠右，一个靠左。回到我们的代码。第一步，先来两个盒子，这个没的说，先来一个div叫做“class = one”,再来一个div叫做 “class = two”，然后呢？给它们加入样式：先给div统一设置宽度200，高度200，紧接着，给.one设置一个背景颜色粉色，给.two也设置一个背景颜色天蓝色，然后按保存，浏览器中查看，不出所料，它们两个一上一下显示，这是行级元素的特点决定的。现在我想让这个盒子就在这里不动，这个盒子跑到这里来，怎么办？

好，按照你们说的来，给div设置display: inline-block，刷新，大家看有没有一左一右？还没有达到预期效果，没关系，我们还有招儿，给.two设置margin-left = 100，够不够？不够继续加，好，可以了吧？现在我把窗口缩小，咦，又到下面去了？能不能用这种方式解决啊？不能。看来目前的知识解决不了这个问题，我们就需要学习新的知识来解决。什么新的知识点呢？就是今天要讲的浮动。

浮动的作用是什么呢？解决一行中显示多个盒子的问题（并且这些盒子的位置是可控的）。

回到代码，把刚才的这几句干掉，再找到.one，给它设置float: left，给two设置float: right，一刷新，走——这个盒子过去没有？并且任意改变浏览器窗口的大小，看这个盒子听不听话？下去的原因是什么？两个盒子的宽度加起来已经超过了浏览器的显示区的宽度，挤不下了。

CSS的 float 属性可以使一个元素脱离正常的文档流，然后被安放到它所在容器的的左端或者右端，并且其他的文本和行内元素环绕它。

当一个元素浮动之后，它会被移出正常的文档流，然后向左或者向右平移，一直平移直到碰到了所处的容器的边框，或者碰到**另外一个浮动的元素**。

一直到浮动元素的margin外侧碰到容器的padding内侧

这个就是浮动，给大家写一下浮动的代码：很简单，就是一个属性：float : left / right / none；简不简单？简单，但是却能实现很多复杂的效果，可以说这个属性的性价比特别高。接下来你们用几分钟的时间自己尝试写一下。

接下来，我们讲一下浮动的特点：

首先呢，先做一个铺垫——叫“标准流”，什么是标准流呢？我们写了这么多html页面，应该有这么一个常识：比如说这是浏览器的显示屏，我先放一个div，它是不是默认靠着左上角？我再放一个div，这个div是不是靠着上面的div在它的下面显示？我再在下面放三个span，这三个是不是在这个div下面在一行中显示出来？下面再来一个div，是不是就在这里显示？像这种正常的布局就称为标准流。标准流就是浏览器默认摆放盒子的标准。

文档流：就是浏览器在页面上摆放元素所用的方法。

对于块级元素，从上到下，如高山流水，一行一行地显示；

对于行级元素，从左到右，

并在每行中按从左至右的挨次排放元素,即为文档流。从上往下，

标准文档流布局（块级元素从上至下，行内元素从左至右）

浮动的第一个特点就是：浮动的元素会脱离标准文档流，在标准流中不占位置了。

什么意思呢？

当一个元素浮动之后，它就像漂浮在空中的气球一样，会脱离正常的文档流，向左或者向右平移，直到碰到了所处容器的padding或者碰到另外一个浮动的元素。

1. 浮动元素脱离了文档流
2. 它后面的块级元素就会当它不存在，浮动以后的元素覆盖在标准流的元素之上
3. 但是文本和行内元素会围绕着它，看起来和word里面的文本环绕的效果一样。
4. 元素浮动之后会导致父元素高度塌陷，因为它的父亲还处在标准文档流的河流中，而浮动元素已经漂到了天上，所以无法撑开父元素了。
5. 浮动元素的外边距不会与其他元素（不论是父元素还是兄弟元素）的外边距合并可以解决外边距折叠的问题Margins of [floating](https://developer.mozilla.org/en-US/docs/Web/CSS/float) and [absolutely positioned](https://developer.mozilla.org/en-US/docs/Web/CSS/position) elements never collapse.
6. 浮动元素的display属性会变成block，比如<span>浮动之后可以设置宽高，<p>浮动之后宽度由内容决定。
7. 浮动找浮动，不浮动（就是标准流中的元素）找不浮动（人以类聚、物以群分）
8. 浮动显示的位置与原本不浮动之前的位置是对应的，这是什么意思呢？我们再新建一个文件04、不浮动的标准流位置会决定浮动以后的位置。如果这个浮动元素有margin，那么margin依然起作用。

现在有三个div，先是都左浮动，显示的是123，全部改成右浮动，显示的是什么？321，为什么会这样呢？因为解析的时候，先找到one，一看它是右浮动，就把每一个元素放到了最右边。接着解析two，它也是右浮动（跑到二楼了），但是二楼最右边的位置是不是已经被人占住了？所以它只能跟在one的后面来显示。three是一样的，它本来应该在最右边的，但是这一块被one占住了，这一块被two占住了，所以它只能靠在这里了。

1. 浮动的元素只会影响下面的元素，不会影响上面的元素。浮动只影响后来者

**float属性会改变元素display属性最终的计算值**。

可以利用浮动的包裹性来达到父容器自适应内部元素宽度。

这里破坏性是指元素浮动后可能导致父元素高度塌陷。

因为浮动元素被从文档正常流中移除了，父元素当然还处在正常流中，所以父元素不能被浮动元素撑大。

浮动元素的外边距不会与其他元素的外边距合并可以解决外边距折叠的问题

我们新建一个页面02-**浮动特性之脱标**

我放三个div，分别设置class，设置宽高： 200，给.one设置背景颜色pink， .two设置天蓝色， .thr设置黄绿色，现在这三个div应该是从上到下依次排列的，没问题吧？现在我给one设置浮动，浮到哪里呢？right，现在推测一下，one会跑到哪里去？two呢？打开浏览器，看和推测一样不一样？one到右边了，two为什么跑到上面去了？这就是浮动的第一个特点： 脱标。即：如果一个元素按照正常的标准流来显示，会在html中所属的位置占位，后面的元素会紧跟着它，但是浮动脱离了标准流，将来在看到浮动的元素之后，不能用正常的标准流来判断。

现在我把one的right改成left，改完之后，页面上好像少了一个元素，另一个到哪里去了呢？按F12打开调试工具，看two被选中的是哪一块？它被one盖住了。

这是第二个特点：**浮动以后的元素覆盖在标准流的元素之上**。two并没有消失，只是被盖在下面了。

好，接着说第三个特点：浮动的规则——**浮动找浮动**，不浮动（就是标准流中的元素）找不浮动（人以类聚、物以群分）。

新建一个页面：03 浮动的规则。现在我把三个盒子全都设置为浮动，出现了什么情况？浮动的元素是不是靠在一起去了？

我在下面再加三个div，这三个有没有浮动？没有。我给它们设置border，现在页面上有几个元素？5个，难道仅仅是看到的这5个吗？不是，还有一个在幕后的，现在可以清楚地看到什么叫浮动找浮动，不浮动找不浮动了吧？

1. 浮动显示的位置与原本不浮动之前的位置是对应的，这是什么意思呢？我们再新建一个文件04、不浮动的标准流位置会决定浮动以后的位置。

现在有三个div，先是都左浮动，显示的是123，全部改成右浮动，显示的是什么？321，为什么会这样呢？因为解析的时候，先找到one，一看它是右浮动，就把每一个元素放到了最右边。接着解析two，它也是右浮动（跑到二楼了），但是二楼最右边的位置是不是已经被人占住了？所以它只能跟在one的后面来显示。three是一样的，它本来应该在最右边的，但是这一块被one占住了，这一块被two占住了，所以它只能靠在这里了。

上面的都讲完了，重点来了——浮动的元素只会影响下面的元素，不会影响上面的元素。

我们再新建一个页面，给它一个05**浮动只影响后来者**。

现在有两个元素，都不浮动，长这个样子。我让第一个浮动，会怎么样？

我让第二个元素浮动，它会不会影响页面的结构？现在把第三个div放出来，它跑哪里去了？

06 判断题：

来5个div，分别加上背景颜色，让2左浮动，4左浮动，

现在大家心里有答案了吗？来看正确答案。

显示的是三个。为什么浮动没有找浮动？因为它们中间还夹了个三，而三是在标准流内的。

如果我现在进行一个切换 ，会发现two / four 是并列的。说明只有在同一个结构中的浮动才会找浮动。

以上都了解了吧？接着说另外一个特点：这个特点也非常重要，是什么呢？叫做：浮动的元素会改变显示方式。

新建一个文件：2个div，两个都是float：left。

不管元素是行内元素还是块级元素，将来显示的时候都会在同一行显示。

我们把div改成span，第一个里面写：今天下雨了，第二个里面写：没有袜子穿了。这时它们在同一行显示，但是设置的宽高有没有生效？你们仔细看啊！看我设置的什么？都左浮动。右键在浏览器中打开，什么情况？是不是有宽有高了？

浮动以后的元素可以设置宽高。总结一下，叫做：**浮动以后的元素的显示方式与行内块级元素一样，既能排成一行也能设置宽高**。

总结一下浮动的几个特征：

浮动的影响：

|  |  |  |  |
| --- | --- | --- | --- |
| 对父元素 | | | 无法撑开其父元素，造成父元素的塌陷 |
| 对兄弟 | 未浮动的 | 块级元素 | 忽视浮动元素的而占据它的位置，并且元素会处在浮动元素的下层（并且无法通过z-index属性改变他们的层叠位置），但它的内部文字和其他行内元素都会环绕浮动元素。 |
| 行级元素 | 环绕浮动元素排列  不想环绕用overflow: hidden |
| 浮动的 | 在左 | 浮动找浮动 |
| 在右 |  |
| 对自身 | | | 变成inline-block |
| 对子元素 | | | 它无法撑开其父元素，但它可以让自己的浮动子元素撑开它自身，并且在没有定义具体宽度情况下，使自身的宽度从100%变为自适应（浮动元素display:block）。其高度和宽度均为浮动元素高度和非浮动元素高度之间的最大值。 |

### 清除浮动

我们平时说的“清除浮动”准确的说应该是“清除浮动造成的影响”，真正的清除浮动应该是float:none;

BFC:块级格式化上下文【在css3中叫Flow Root】是一个独立布局环境，相邻盒子margin垂直方向会重叠。

什么样的元素会为其内容生成一个BFC呢？

* 浮动元素，即float:left|right
* 绝对定位元素，即position:absolute|fixed
* 块容器【block containers】(比如inline-blocks，table-cells和table-captions)，这些块容器有个特征就是它们不是块级盒子【block boxes】。即display:table-cell|table-caption|inline-block
* 设置了除visible外的overflow值的块盒子【blok boxes】，即overflow:hidden|scroll|auto

BFC特性：

* 创建了BFC的元素中，子浮动元素也会参与高度计算
* 与浮动元素相邻的、创建了BFC的元素，都不能与浮动元素相互覆盖
* 创建了BFC的元素不会与它们的子元素margin重叠