# CSS基础入门

第4天课堂笔记（本课程共8天）

# 目录

[CSS基础入门 1](#_Toc16644)

[目录 2](#_Toc9147)

[一、 复习 3](#_Toc19959)

[二、 浮动的性质 3](#_Toc18574)

[1、 浮动的元素脱离标准流 4](#_Toc23444)

[2、 浮动的元素依次贴边 4](#_Toc3598)

[3、浮动的元素没有margin塌陷 5](#_Toc6191)

[4、浮动的元素让出标准流的位置 5](#_Toc10384)

[5、 字围 6](#_Toc28877)

[三、 清除浮动 7](#_Toc23272)

[1、浮动存在的问题。 7](#_Toc1872)

[浮动的盒子不能撑高父亲。 7](#_Toc26722)

[浮动的元素会影响后面的浮动元素。 7](#_Toc24313)

[2、 方法一：给盒子添加高度。 7](#_Toc1979)

[3、 方法二：给受影响的盒子添加clear属性 7](#_Toc6673)

[4、方法三：隔墙法。 8](#_Toc26284)

[5、 方法四：overflow：hidden 9](#_Toc14772)

# 复习

1. margin

margin塌陷：在标准流中，相邻的两个盒子，垂直方向，上面的有下边距，下面的有上边距，实际的距离是两个边距之间最大的，并不会进行叠加。

儿子不能踹父亲：如果子级和父级之间有一个间距，这个间距要用父级的padding去撑，而不是用子级的margin去撑。

子盒子在父盒子内水平居中：

|  |
| --- |
| 1. width: 900px; 2. margin: 0 auto; |

文字水平居中：

|  |
| --- |
| 1. text-align: center; |

文字单行垂直居中：让行高等于盒子高。

文字多行、盒子、其他内容垂直居中：父盒子不给高度，用内部元素内容撑高，给父盒子一个上下相同的padding。

1. 宽度剩余：如果右边剩余的空间不同，可以给大盒子一个足够的宽度。
2. height：有的情况下，高度用内容撑高，不是设定死的。
3. 父子盒模型：儿子的整体占位不能超过父级的内容区域。
4. 标准文档流：网页制作的时候，是从上往下依次制作。

块级元素：div、p、h系列、ul、ol、li、dl、dt、dd等。

行内元素：span、img、a、input等。

块级元素性质：

独占一行；设置宽高；没有宽度会自动撑满父亲。

行内元素性质：

排列在一行；不能设置宽高；宽高是用内容撑开的。

可以相互切换：display显示模式。

|  |
| --- |
| 1. display: block; 常用 2. display: inline; |

真正制作网页，需要让元素脱离标准流。

脱标：元素既可以排在一行，又可以设置宽高。

脱标的三种方式：浮动、绝对定位、固定定位。

浮动：让元素排在一行，便于布局。

float：属性值left、right。

# 浮动的性质

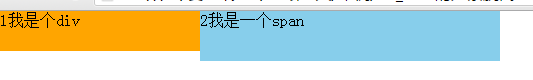
浮动：用来制作网页布局的最主要属性。

## 浮动的元素脱离标准流

浮动的元素不再是我们标准流中的block或inline之分。本身既可以设置宽高，又能多个浮动的元素排在一行。

浮动的元素就脱离标准流。

|  |
| --- |
| 1. float：left; |



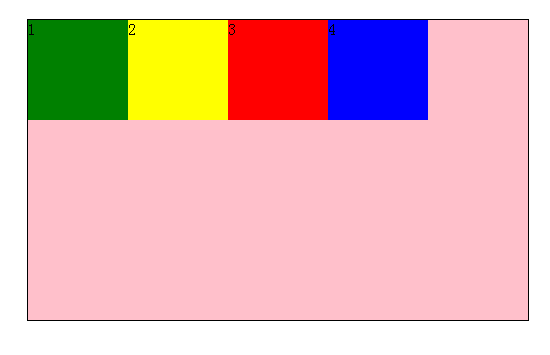
可以利用浮动的这个特性：让我们想要的元素排在一行。

## 浮动的元素依次贴边

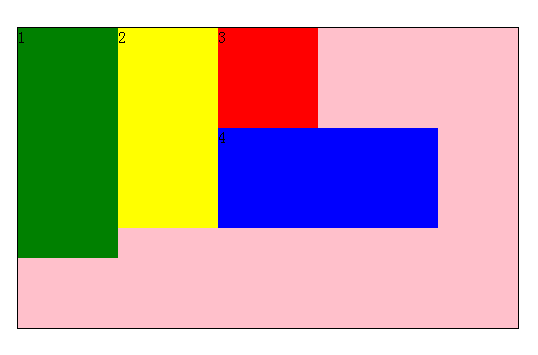
贴边的元素有方向之分。

同一个方向的浮动的元素贴边，以左边为例：

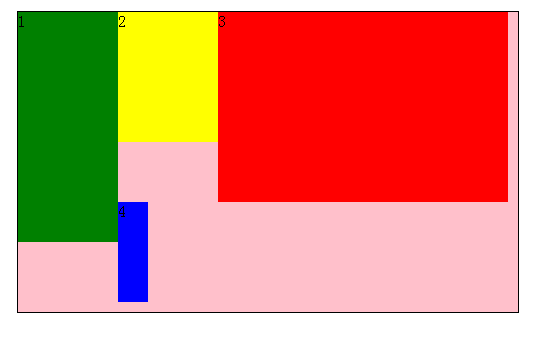
父盒子的左边框内部 ← 子盒子1 ← 子盒子2 ← 子盒子3 ← 子盒子4……



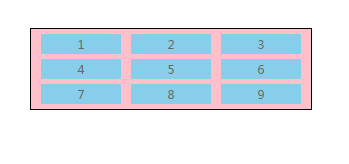
如果前面的盒子留的距离不够了， 向前一个继续去贴边。



如果要贴的盒子中间有空隙，它不会去钻盒子。

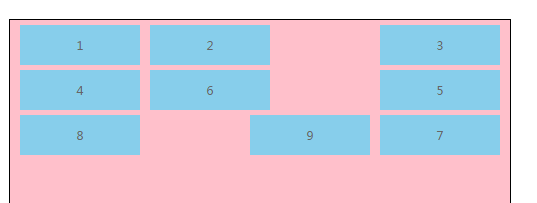


案例：制作表格列表项。



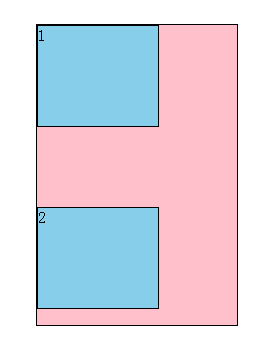
如果同一个大盒子里有两个方向浮动的元素，会找同一方向的前一个元素去贴边。

|  |
| --- |
| 1. <ul> 2. <li>1</li> 3. <li>2</li> 4. <li class="right">3</li> 5. <li>4</li> 6. <li class="right">5</li> 7. <li>6</li> 8. <li class="right">7</li> 9. <li>8</li> 10. <li class="right">9</li> 11. </ul> |



## 3、浮动的元素没有margin塌陷

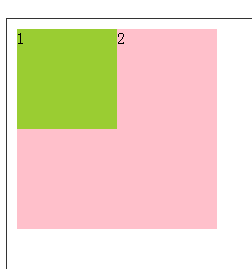
标准流里有margin塌陷现象。



## 4、浮动的元素让出标准流的位置

同一级兄弟元素，前面的元素浮动，后面的元素没有浮动。那么浮动元素会让出标准流的位置给后面的元素。

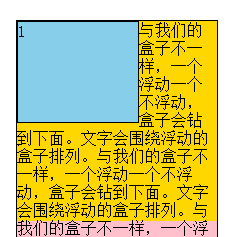
|  |
| --- |
| 1. .box ul .item1{ 2. float: left; 3. width: 100px; 4. height: 100px; 5. background: yellowgreen; 6. } 7. .box ul .item2{ 8. width: 200px; 9. height: 200px; 10. background: pink; 11. } |



视觉效果上有一个压盖效果，我们实际工作中，不用浮动做压盖。我们用定位来制作压盖效果。

## 字围

与我们的盒子不一样，一个浮动一个不浮动，盒子会钻到下面。文字会围绕浮动的盒子排列。



案例：宋小宝简介。字围效果。

|  |
| --- |
| 1. .box img{ 2. float: left; 3. width: 150px; 4. height: 200px; 5. margin: 0 10px 10px 0; 6. } 7. .box p{ 8. font-size: 16px; 9. line-height: 30px; 10. text-indent: 2em; 11. } |

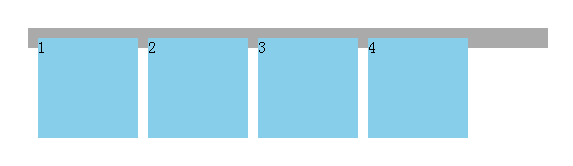


# 清除浮动

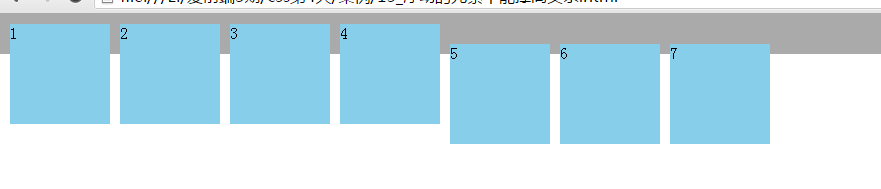
## 1、浮动存在的问题。

### 浮动的盒子不能撑高父亲。

|  |
| --- |
| 1. .box p{ 2. float: left; 3. width: 100px; 4. height: 100px; 5. margin-right: 10px; 6. background: skyblue; 7. } |



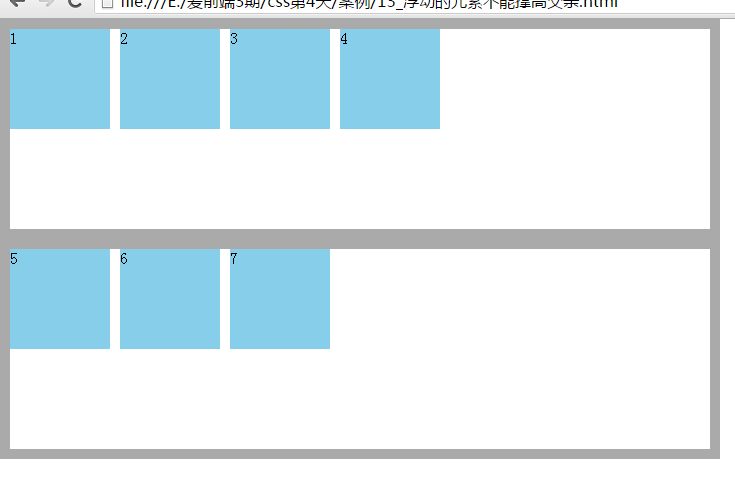
### 浮动的元素会影响后面的浮动元素。



我们需要解决问题：

## 方法一：给盒子添加高度。

最直接有效的方法：有高度的盒子能够管住他内部的浮动的元素。不受其他浮动影响，也不去影响其他元素。



存在一个问题：高度不能自适应。浮动的元素还是不能撑高盒子。

## 方法二：给受影响的盒子添加clear属性

clear：清除。

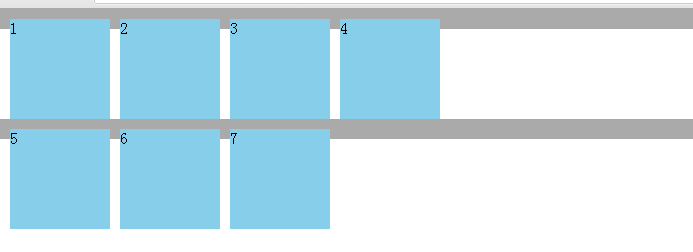
三个属性值：left、right、both。

left：清除元素受到的左浮动的影响。

right：清除元素受到的右浮动的影响。

both：清除两边的影响。

|  |
| --- |
| 1. clear:both; |



解决：浮动互相影响。

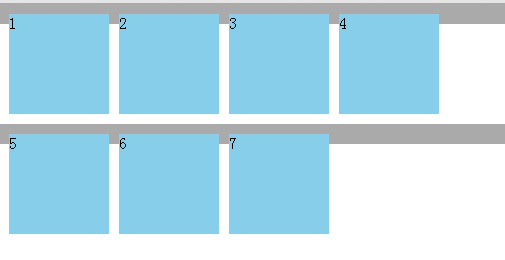
没解决：高度自适应。margin显示效果失效。

|  |
| --- |
| 1. .box{ 2. width: 700px; 3. border: 10px solid #aaa; 4. margin-bottom: 10px; //margin的值必须大于子盒子的最高才能看出边距 5. } |

## 4、方法三：隔墙法。

外墙法：在互相影响的父盒子之间建一堵墙。这个墙给它一个高度，让它有个属性叫做clear。

|  |
| --- |
| 1. <div class="cl h10"></div> |



解决：浮动互相影响。视觉上有了间距。

没解决：高度自适应。实际的margin问题还存在。

内墙法：

将清除了浮动的墙建在有浮动的父盒子的最底部。

|  |
| --- |
| 1. <div class="box"> 2. <p>1</p> 3. <p>2</p> 4. <p>3</p> 5. <p>4</p> 6. <div class="cl"></div> 7. </div> |



解决：浮动影响。高度自适应。margin失效。

方法很好，但是还是不太常用。因为我们添加了很多没有语义的div。对我的整体结构看起来比较臃肿。

## 方法四：overflow：hidden

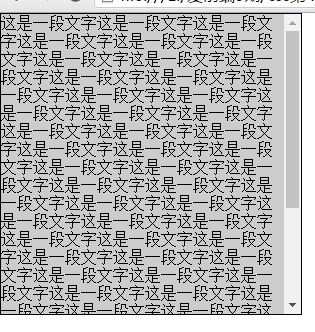
认识overflow：溢出处理方法。

两个属性值：auto和hidden。

溢出滚动：

如果盒子里面的内容超过我设置的高度，会出现滚动条。显示的只有高度部分，拉动滚动条可以查看全部内容。

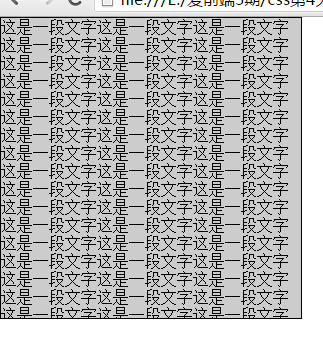
|  |
| --- |
| 1. overflow: auto; |



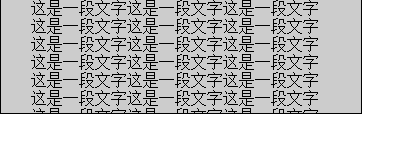
溢出隐藏：

超出盒子的部分，自动隐藏起来。

|  |
| --- |
| 1. overflow：hidden； |



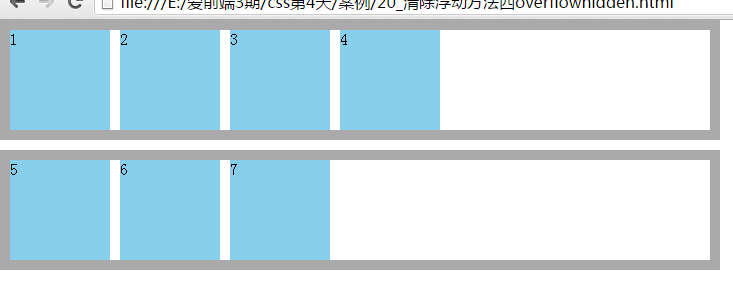
注意：溢出隐藏，对于下padding的部分并不能隐藏，隐藏border边框外面的部分。



清除浮动：能用溢出隐藏属性解决。原理我们不必深究。

解决：浮动影响。高度自适应。margin失效。

|  |
| --- |
| 1. .box1{ 里面的元素浮动了，加溢出隐藏直接解决所有问题。 2. width: 700px; 3. border: 10px solid #aaa; 4. overflow: hidden; 5. } |



overflow：hidden属性的两个作用：

强制我们的父盒子去检测子盒子的高度。让父盒子的高度用检测的高度撑高。（以后碰到让盒子高度自适应，一定要给盒子加一个overflow：hidden）。

强制我们的父盒子去检测浮动的元素。将这些元素管理起来，不让他影响别人。

清除浮动最简单，效果也是最好的。

实际工作：

1. 我们只要碰到盒子里面有浮动的元素，就给她加overflow：hidden。
2. 一般会在最大的几个板块之间，隔一道外墙。

案例：tab栏（选项卡）