# CSS入门和高级技巧

第4天课堂笔记（本课程共8天）

目录

[CSS入门和高级技巧 1](#_Toc17949)

[目录 2](#_Toc2500)

[一、复习 3](#_Toc8426)

[二、深入了解浮动的性质 5](#_Toc20827)

[2.1 浮动的元素脱离标准流，没有标准流的行块之分了 5](#_Toc18893)

[2.2 依次贴边 6](#_Toc16603)

[2.3 竖直方向上的margin塌陷现象消失 7](#_Toc371)

[2.4 让出了标准流 7](#_Toc14742)

[2.5 字围 8](#_Toc8662)

[三、清除浮动 9](#_Toc31692)

[3.1 父亲不能被浮动的儿子撑出高 9](#_Toc10175)

[3.2 治这个病的一个偏方 10](#_Toc24727)

[3.3 有高度的盒子能够管住自己内部的浮动元素 12](#_Toc360)

[3.4 清除浮动方法1：clear:both; 13](#_Toc30869)

[3.5 清除浮动方法2：隔墙 14](#_Toc4898)

[3.6 清除浮动方法3：内墙法 15](#_Toc23660)

[3.7 清除浮动方法4：overflow:hidden;法 15](#_Toc10792)

# 一、复习

● CSS ： 负责样式层，层叠式样式表cascading style sheet。CSS2.1，最新版本CSS3。

● CSS选择器： 选择哪些元素加样式。基本选择3种：标签p、id选择器#id、class选择器.；高级选择器4种：后代选择器div p 、交集选择器div.haha 、并集选择器div,p 、通配符\*

id选择器：全页面id唯一；

class选择器：同一个标签可以带多个class：

|  |
| --- |
| 1. <p class=”para1 spec”></p> |

class选择器不要求页面唯一

|  |
| --- |
| 1. <p class=”para1 spec”></p> 2. <h3 class=”spec”></h3> |

● 继承性：

哪些属性能继承：color、font-、text-、line-开头的。

不能继承的：background-color、盒模型的属性、浮动、定位

|  |
| --- |
| 1. body{ 2. font-size:12px; 3. } |

● 层叠性：

遇见冲突，听谁的。有一个非常严密的图。!important提升权重，要知道!important能提升什么，不能提升什么？比如不能影响就近原则：离得近的就是近的，远的写!important没用；不能把继承来的提升权重：选中了的就是选中了的，继承来的!important也无法干掉选中了的。

● 文本属性：

|  |
| --- |
| 1. font-weight:bold; 2. font-weight:700; 3. font-style:italic; 4. text-decoration:underline; 5. font-size:12px; 6. line-height:24px; 7. font-family:”Consolas”,”Microsoft Yahei”,”SimSun”; 8. font:12px/24px ”Consolas”,”Microsoft Yahei”,”SimSun”; |

● 盒模型

|  |
| --- |
| 1. width:200px; /\*盒子内容的宽度\*/ 2. height:200px; /\*盒子内容的高度\*/ 3. padding:10px; /\*内边距\*/ 4. border:1px solid red; 5. margin:40px; |

我们在这里补充强调一下：padding区域有背景颜色；并且现在的知识水平，你不能给padding

区域单独设置颜色；

|  |
| --- |
| 1. padding: 10px 20px 30px 40px; |

上、右、下、左

|  |
| --- |
| 1. padding:10px 20px 30px; |

上、左右、下

边框的三要素：

|  |
| --- |
| 1. border-width:1px; 2. border-style:solid; 3. border-color:red; |

还能继续拆：

|  |
| --- |
| 1. border-top-style:dashed; |

三要素的写法可以上右下左：

|  |
| --- |
| 1. border-width:1px 2px 3px 4px; |

● 浮动：我们还没有科学的介绍，一些性质还不知道，没关系作业已经能做了。因为浮动是干嘛的我们已经知道了：制作并排的。

标准文档流中元素分为两种：

block-level ： 能够设置宽高、不能并排，比如div、h、p、li、dt、dd

inline-level： 不能设置宽高、能并排，比如span、a、b、u、i

浮动，就能让元素又能并排，又能设置宽高。

# 二、深入了解浮动的性质

## 2.1 浮动的元素脱离标准流，没有标准流的行块之分了

脱离标准流一共就3个方法：

浮动：

|  |
| --- |
| 1. float:left; |

绝对定位：

|  |
| --- |
| 1. position:absolute; |

固定定位：

|  |
| --- |
| 1. position:fixed; |

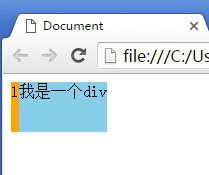
浮动的元素已经脱离了标准流，所以没有inline、block之分了。

span在标准流中一个经典的行内元素，但是浮动了，就可以不转块直接设置宽度、高度。

|  |
| --- |
| 1. <span class="no1">1</span> |

|  |
| --- |
| 1. .no1{ 2. **float: left;** 3. width: 300px; 4. height: 50px; 5. background-color: orange; 6. } |

div在标准流中是一个经典的块级元素，不设置宽度在标准流中是自动撑满父亲的width。**但是浮动了，就不自动撑满了，而是自动收缩了，收缩为内部文字的大小了**：

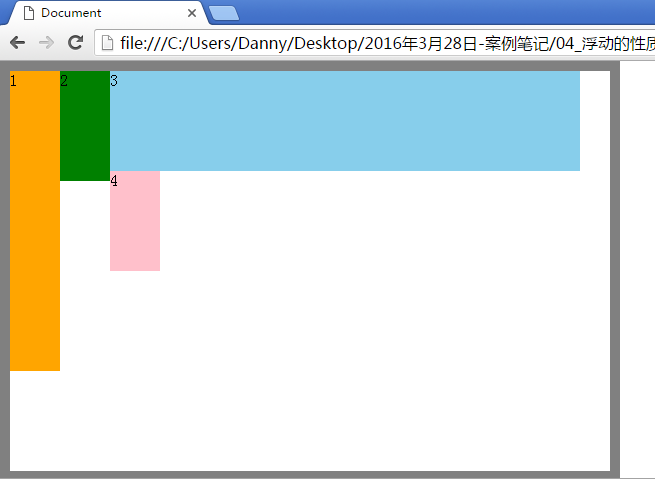


## 2.2 依次贴边

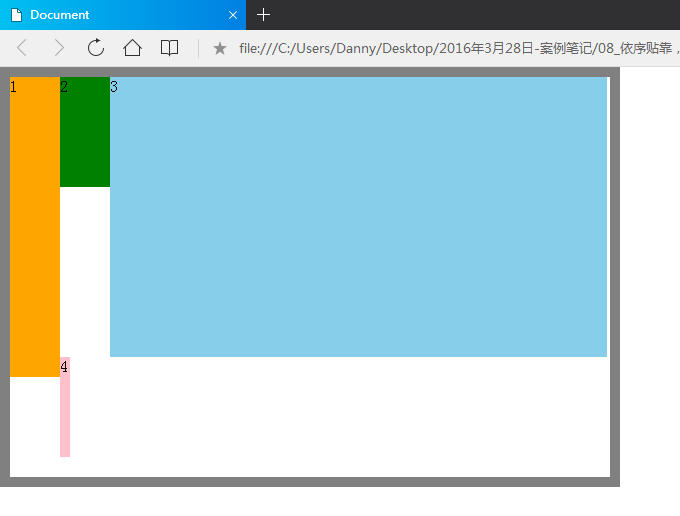
左浮动：

父盒子的左边框内部 ← 老1 ← 老2 ← 老3 ← 老4

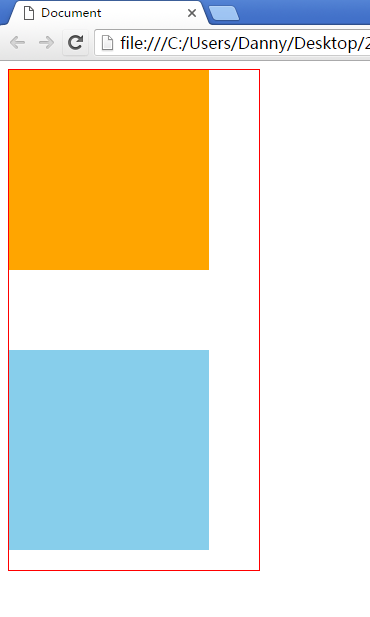
如果之前的兄弟已经不足以容纳自己，比如老4没有足够的空间并排了，那么将依次寻找老3、老2、老1、父亲的边框去贴：



但是不钻：



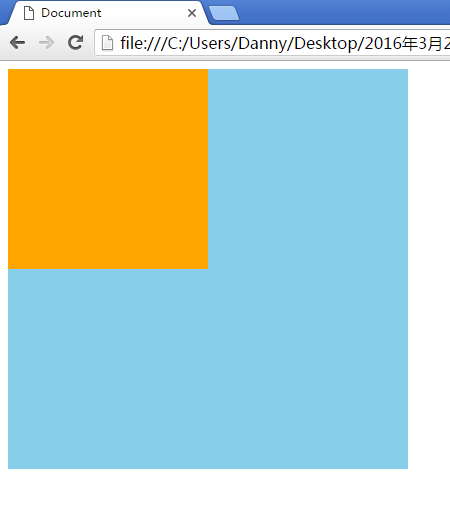
## 2.3 竖直方向上的margin塌陷现象消失



margin塌陷是标准流的性质，因为浮动脱标了，就没有这个事儿了。

## 2.4 让出了标准流

|  |
| --- |
| 1. <div class="box1"></div> → 浮动 2. <div class="box2"></div> → 没有浮动 |



**注意，这个性质没啥用，工作中制作“压盖”使用定位，而不是用这个浮动的小技巧。**

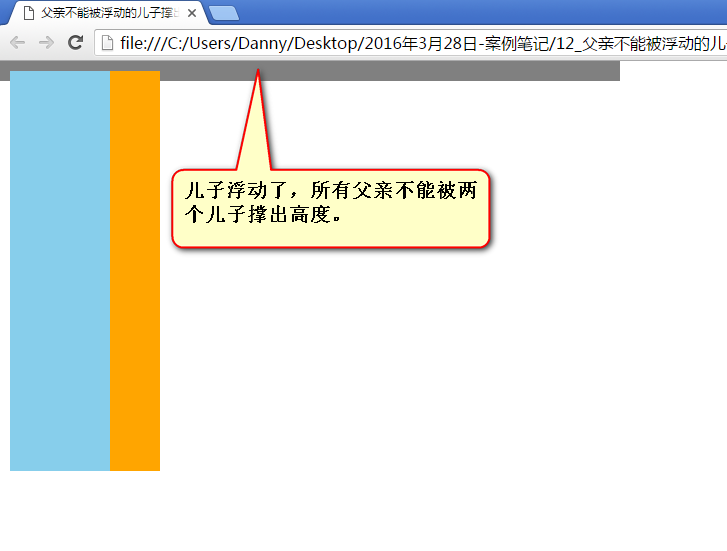
橙色盒子浮动了，让出了标准流的位置，标准流的光标还在页面左上角，所以蓝色盒子就渲染在左上角，被橙色盒子压住了。

## 2.5 字围



# 三、清除浮动

## 3.1 父亲不能被浮动的儿子撑出高



下面的代码，no1、no2都浮动了，box就不能被儿子撑出高度：

|  |
| --- |
| 1. <div class="box"> 2. <p class="no1"></p> 3. <p class="no2"></p> 4. </div> |

因为父亲只能被标准流的元素撑高。

## 3.2 治这个病的一个偏方

有一个属性叫做

|  |
| --- |
| 1. overflow:hidden; |

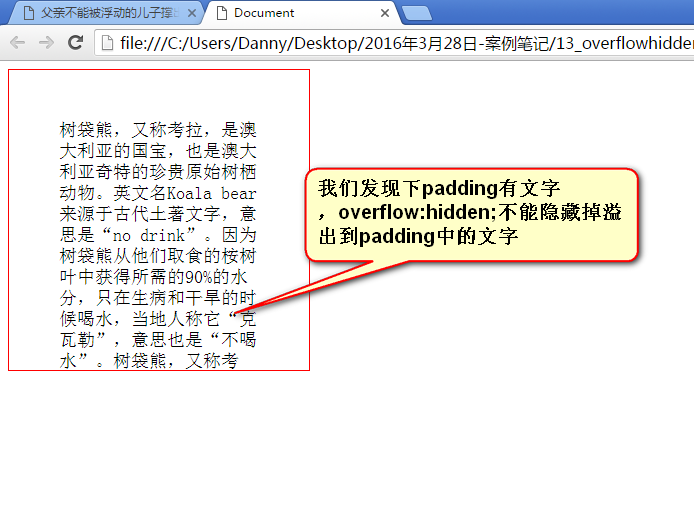
能够解决事情。

现在先来学习overflow:hidden;的本意是什么

|  |  |
| --- | --- |
|  |  |

overflow是英语“溢出”的意思；hidden就是隐藏的意思。

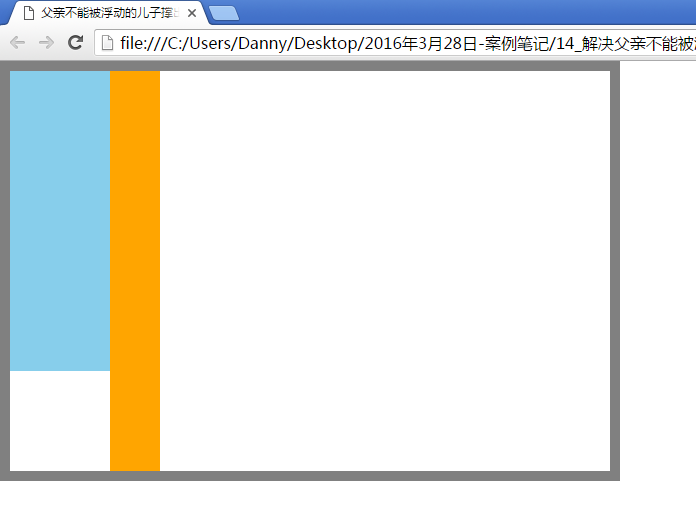
这个属性的意思，就是让溢出边框的内容隐藏。



就如同橘子皮就是橘子皮，但是我们发现可以治感冒，就是世界上存在很多这样的事情，八竿子打不着的事儿，居然有联系。

overflow:hidden;是用来隐藏掉溢出边框的内容的，是用来变魔术的。但是我们发现，overflow:hidden;有神奇的别的用处，就是能够让父亲认识自己脱标的儿子，能够让父亲被自己脱标的儿子撑出高度。

|  |
| --- |
| 1. <div class="box"> →不能被撑出高了，解决办法： **overflow:hidden;** 2. <p class="no1"></p> → 脱标了 3. <p class="no2"></p> → 脱标了 4. </div> |



## 3.3 有高度的盒子能够管住自己内部的浮动元素

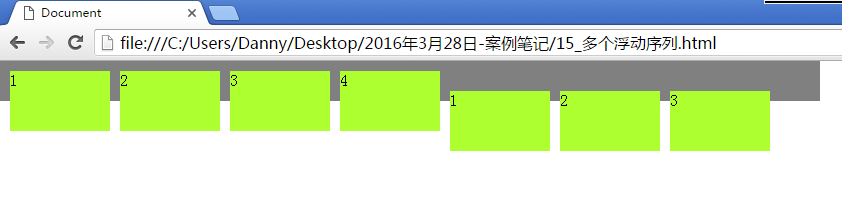


有高度的盒子，能够管住自己的内部的浮动元素，不会影响别人内部的浮动元素，也不会受别人影响。

|  |
| --- |
| 1. <div class="box1"> 2. <p>1</p> 3. <p>2</p> 4. <p>3</p> 5. <p>4</p> 6. </div> 7. <div class="box2"> 8. <p>1</p> 9. <p>2</p> 10. <p>3</p> 11. </div> |

## 3.4 清除浮动方法1：clear:both;

我们发现，有高度的盒子可以管住自己的儿子，但是如果父盒子没有高度，那么序列就乱套了，第二个序列就去追随第一个序列了：



解决办法挺简单，就是给后面的盒子加上

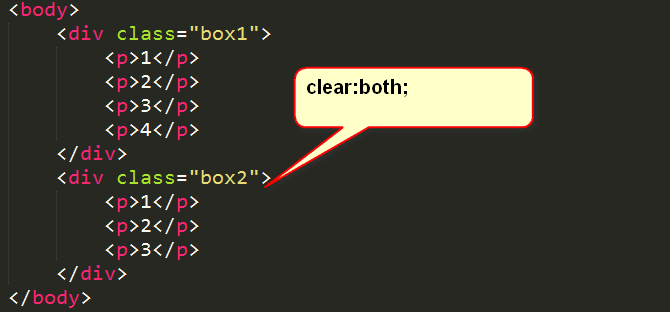
|  |
| --- |
| 1. clear:both; |

clear就是清除意思，它的值可以是：

clear:left; 表示清除左浮动带来的影响

clear:right; 表示清除右浮动带来的影响

clear:both；表示清除所有浮动带来的影响



这个东西不好用，原因是：

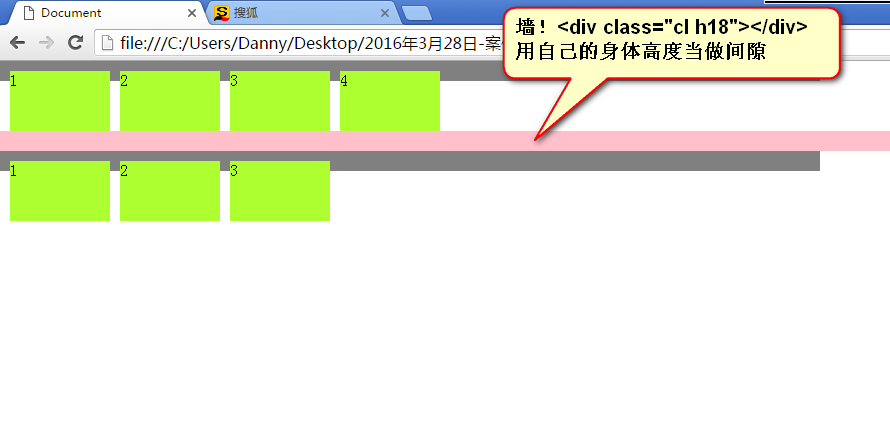
1） box还是没有高

2） margin失效

## 3.5 清除浮动方法2：隔墙

隔墙法是非常常用的，在网页中两个非常大的部分，我们总希望之间隔一堵墙，把两部分内部的浮动都关在里面，不要相互影响，小技巧就是margin依旧失效，可以用墙的高度来模拟间隔。

|  |
| --- |
| 1. <div class="box1"> 2. <p>1</p> 3. <p>2</p> 4. <p>3</p> 5. <p>4</p> 6. </div> 7. **<div class="cl h20"></div>** 8. <div class="box2"> 9. <p>1</p> 10. <p>2</p> 11. <p>3</p> 12. </div> |



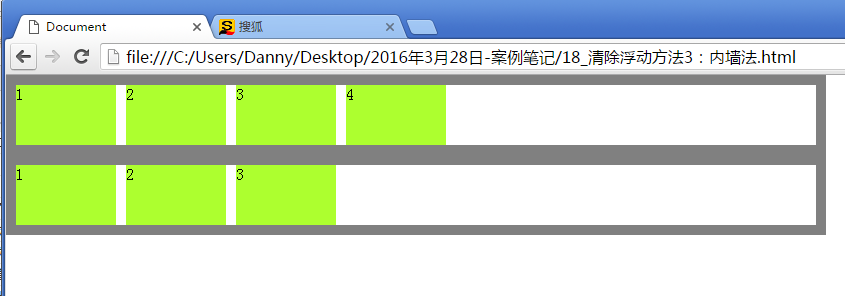
margin虽然失效，但是可以用小技巧来弥补，可以用墙的高来当做间隔。

但是盒子依然没有高。

## 3.6 清除浮动方法3：内墙法

cl就是clear:both;

|  |
| --- |
| 1. <div class="box1"> 2. <p>1</p> 3. <p>2</p> 4. <p>3</p> 5. <p>4</p> 6. **<div class="cl"></div>** 7. </div> 9. <div class="box2"> 10. <p>1</p> 11. <p>2</p> 12. <p>3</p> 13. **<div class="cl"></div>** 14. </div> |



现在margin好用了，并且盒子也有高了，所有的事情都解决了！但是还是有问题：

HTML标签有点放置的太多了。这些标签页没有语义，看起来不爽。

## 3.7 清除浮动方法4：overflow:hidden;法

|  |
| --- |
| 1. <div class="box1"> **→ 这个盒子没高，就写上overflow:hidden;** 2. <p>1</p> 3. <p>2</p> 4. <p>3</p> 5. <p>4</p> 6. </div> 8. <div class="box2"> **→ 这个盒子没高，就写上overflow:hidden;** 9. <p>1</p> 10. <p>2</p> 11. <p>3</p> 12. </div> |

总结：记住，清除浮动的方法有很多，但是很多都是在面试中有意义。工作中：

**1） 我们总喜欢给内部有浮动的父盒子加上overflow:hidden;**

**2) 我们总喜欢在两个大部分之间隔外墙**

|  |
| --- |
| 1. <div class="header"> 3. </div> 4. **<div class="cl h18"></div>** 5. <div class="content"> 7. </div> 8. **<div class="cl h18"></div>** 9. <div class="footer"> 11. </div> |