第2周课堂笔记

HLH

# 目录

[一、盒模型 3](#_Toc81955499)

[1.1 盒子中的区域 3](#_Toc81955500)

[二、标准文档流 4](#_Toc81955501)

[2.1 块级元素和行内元素 4](#_Toc81955502)

[2.2 块级元素和行内元素的相互转换 4](#_Toc81955503)

[三、浮动的清除 5](#_Toc81955504)

[3.1 清除浮动方法1：给浮动的元素的祖先元素加高度。 5](#_Toc81955505)

[3.2清除浮动方法3：内墙法; 5](#_Toc81955506)

[3.3 清除浮动方法3：overflow:hidden; 6](#_Toc81955507)

[3.4 伪元素（用的最多，一定记住.clear2 按tab修改） 6](#_Toc81955508)

[3.5 双伪元素（.clear1 按tab修改） 7](#_Toc81955509)

[四、margin 7](#_Toc81955510)

[4.1 margin的塌陷现象 7](#_Toc81955511)

[4.2 盒子居中margin:0 auto; 8](#_Toc81955512)

[4.3 外边距塌陷 8](#_Toc81955513)

[4.4 关于margin的IE6兼容问题 9](#_Toc81955514)

# 一、盒模型

## 1.1 盒子中的区域

一个盒子中主要的属性就5个：width、height、padding、border、margin。

width是“宽度”的意思，CSS中width指的是内容的宽度，而不是盒子的宽度。

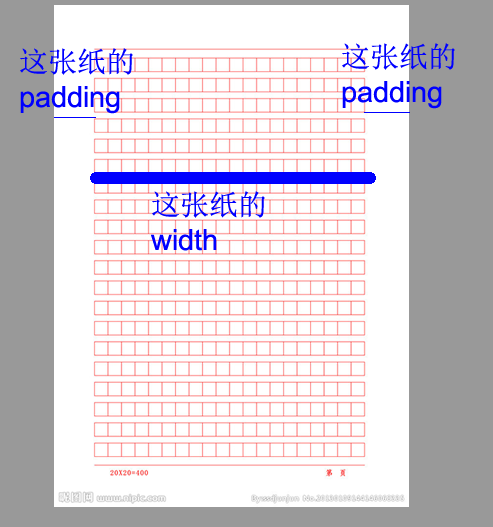
height是“高度”的意思，CSS中height指的是内容的高度，而不是盒子的高度

padding是“内边距”的意思

border是“边框”

margin是“外边距”

盒模型的示意图：



下面这两个盒子，真实占有宽高，完全相同，都是302\*302：

真实占有宽度= 左border + 左padding + width + 右padding + 右border

这两个盒子的盒模型图，见下表：

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

这三个盒子的盒模型图：

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

如果想保持一个盒子的真实占有宽度不变，那么加width就要减padding。加padding就要减width。

# 二、标准文档流

## 2.1 块级元素和行内元素

标准文档流等级森严。标签分为两种等级：

1） 块级元素

● 霸占一行，不能与其他任何元素并列

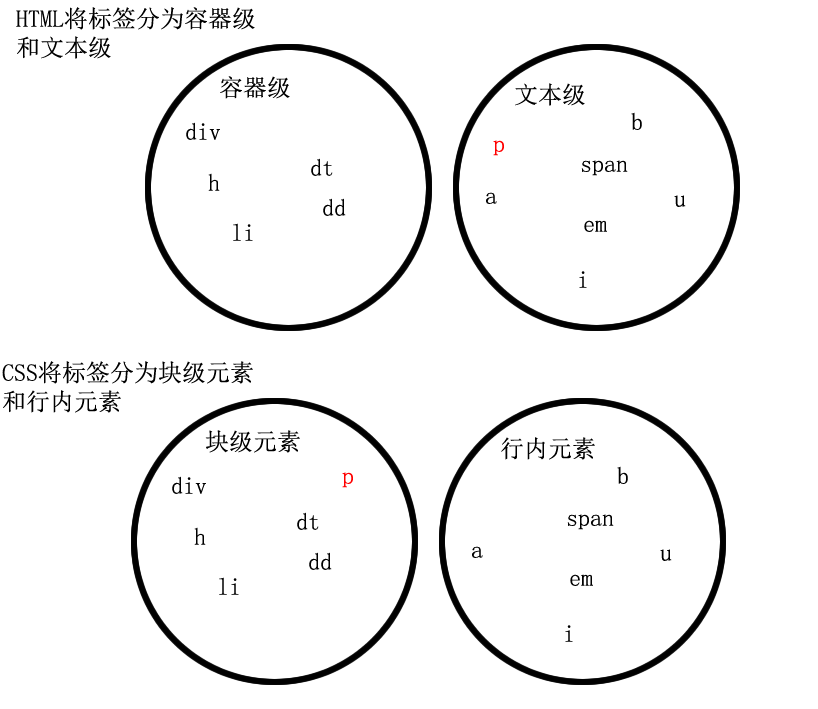
● 能接受宽、高

● 如果不设置宽度，那么宽度将默认变为父亲的100%。

2） 行内元素

● 与其他行内元素并排

● 不能设置宽、高。默认的宽度，就是文字的宽度。



## 2.2 块级元素和行内元素的相互转换

块级元素可以设置为行内元素

行内元素可以设置为块级元素

|  |
| --- |
| 1. div{ 2. display: inline; 3. background-color: pink; 4. width: 500px; 5. height: 500px; 6. } |

display是“显示模式”的意思，用来改变元素的行内、块级性质

inline就是“行内”。

一旦，给一个标签设置

|  |
| --- |
| 1. display: inline; |

那么，这个标签将立即变为行内元素。此时它和一个span无异：

● 此时这个div设置宽度、高度无效；

● 此时这个div可以和别人并排了。

同样的道理，

|  |
| --- |
| 1. span{ 2. display: block; 3. width: 200px; 4. height: 200px; 5. background-color: pink; 6. } |

“block”是“块”的意思

让标签变为块级元素。

● 此时这个span能够设置宽度、高度

● 此时这个span必须霸占一行了，别人无法和他并排

● 如果不设置宽度，将撑满父亲

如果我们现在就要块级元素并排、并且要设置宽高？

脱离标准流！

css中一共有三种手段，使一个元素脱离标准文档流：

1） 浮动

2） 绝对定位

3） 固定定位

一旦一个元素浮动了，那么，将能够并排了，并且能够设置宽高了。无论它原来是个div还是个span。

|  |
| --- |
| 1. span{ 2. float: left; 3. width: 200px; 4. height: 200px; 5. background-color: orange; 6. } |

# 三、浮动的清除

原因就是因为div没有高度，不能给自己浮动的孩子们，一个容器。

## 3.1 清除浮动方法1：给浮动的元素的祖先元素加高度。

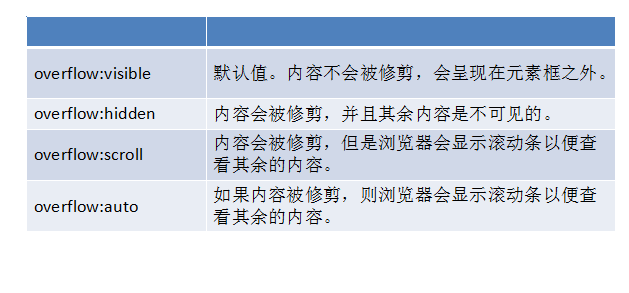
**高度的盒子，能够关住浮动。**

缺点：不灵活

## 3.2清除浮动方法3：内墙法;

clear:both;

## 3.3 清除浮动方法3：overflow:hidden;



overflow就是“溢出”的意思， hidden就是“隐藏”的意思。

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

表示“溢出隐藏”。所有溢出边框的内容，都要隐藏掉。

|  |  |
| --- | --- |
| 内容太多，溢出了盒子： | overflow:hidden; 溢出盒子边框的内容，隐藏了。 |

本意就是清除溢出到盒子外面的文字。

## 3.4 伪元素（用的最多，一定记住.clear2 按tab修改）

伪元素:

.clearfix:after {

content:””; //也可以用content:”.”;

visibility:hidden;

display:block;

height:0;

clear:both;

}

.clearfix{

Zoom:1;

}

## 3.5 双伪元素（.clear1 按tab修改）

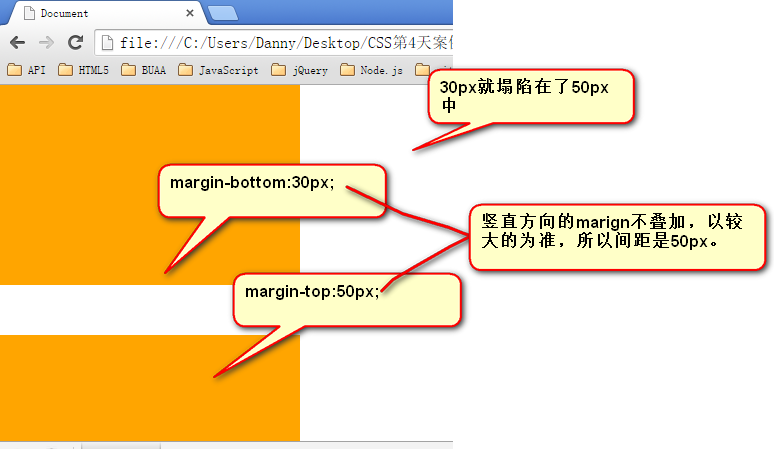
双伪元素

.clearfix:before,.clearfix:after{  
 display: table;  
 content: "";  
}  
.clearfix:after {  
 clear: both;  
}  
.clearfix {  
 zoom: 1;  
}

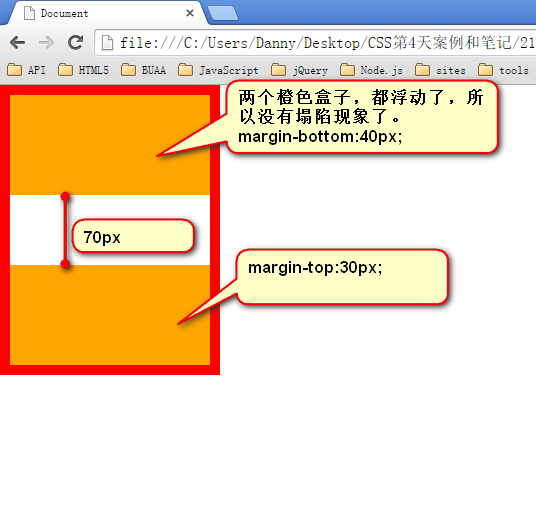
# 四、margin

## 4.1 margin的塌陷现象

**标准文档流中，竖直方向的margin不叠加，以较大的为准。**



如果不在标准流，比如盒子都浮动了，那么两个盒子之间是没有塌陷现象的：



## 4.2 盒子居中margin:0 auto;

margin的值可以为auto，表示自动。当left、right两个方向，都是auto的时候，盒子居中了：

简写为

|  |
| --- |
| 1. margin:0 auto; |

注意：

1） 使用margin:0 auto; 的盒子，必须有width，有明确的width

2） 只有标准流的盒子，才能使用margin:0 auto; 居中。

也就是说，当一个盒子浮动了、绝对定位了、固定定位了，都不能使用margin:0 auto;

3） margin:0 auto;是在居中盒子，不是居中文本。

文本的居中，要使用

|  |
| --- |
| 1. text-align:center; |
| 1. margin:0 auto; → 让这个div自己在大容器中居中。 2. text-align: center; → 让这个div内部的文本居中。 |

## 4.3 外边距塌陷

这个son有一个margin-top踹父亲，试图将自己下移

|  |
| --- |
| 1. <div class=”father”> 2. <div class=”son”></div> 3. </div> |

结果：

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

**margin这个属性，本质上描述的是兄弟和兄弟之间的距离。**

解决方案:

◆给父盒子加border

◆给父盒子加overflow:hidden;

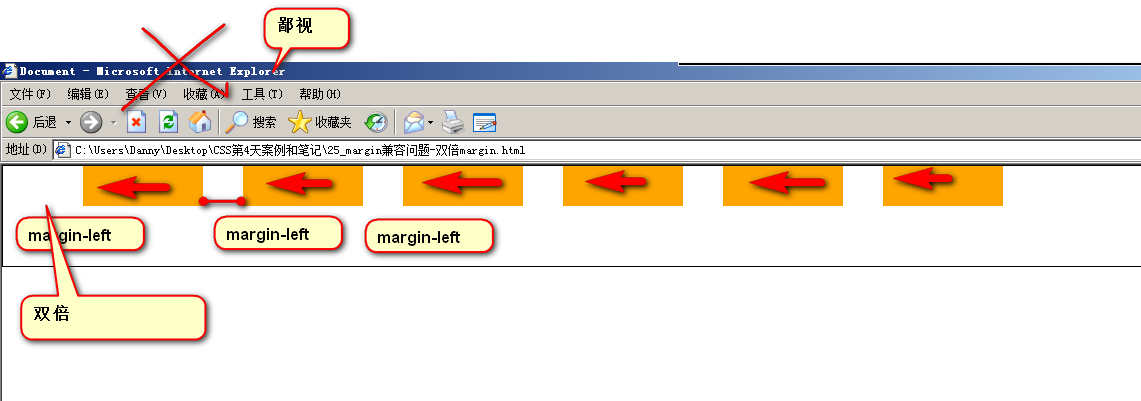
父子间，属于内边矩。用padding

## 4.4 关于margin的IE6兼容问题

IE6双倍margin bug

**当出现连续浮动的元素，携带和浮动方向相同的margin时，队首的元素，会双倍marign。**

|  |
| --- |
| 1. <ul> 2. <li></li> 3. <li></li> 4. <li></li> 5. </ul> |



解决方案：

1）使浮动的方向和margin的方向，相反。左浮动，右margin.

浮动的方向和margin的方向相反。把这个当做习惯去做。

|  |
| --- |
| 1. float: left; 2. margin-right: 40px; |