# CSS入门和高级技巧

第3天课堂笔记（本课程共8天）

目录

[CSS入门和高级技巧](#_Toc20372)

[目录](#_Toc29167)

[一、上节课知识复习](#_Toc28205)

[第1题 会审查元素](#_Toc24048)

[第2题 颜色表示](#_Toc30176)

[第3题 选择器权重](#_Toc31321)

[第4题 选择器权重](#_Toc16964)

[第5题 选择器权重](#_Toc31729)

[第6题 选择器权重](#_Toc9804)

[第7题 选择器权重](#_Toc15875)

[二、盒模型](#_Toc7092)

[2.1 会看懂盒模型图](#_Toc6287)

[2.2 padding内边距](#_Toc4841)

[2.3 border边框](#_Toc30990)

[2.4 margin外边距](#_Toc3889)

[三、居中策略](#_Toc23167)

[3.1 文字水平居中](#_Toc22816)

[3.2 盒子水平居中](#_Toc24202)

[3.3 单行文本的垂直居中](#_Toc8602)

[四、标准文档流](#_Toc6390)

[4.1 什么是标准文档流](#_Toc492)

[4.2 标准文档流有什么性质](#_Toc26587)

[4.2.1 空白折叠现象](#_Toc27085)

[4.2.2 高矮不齐，底边对齐](#_Toc17956)

[4.3 块级元素和行内级元素](#_Toc21062)

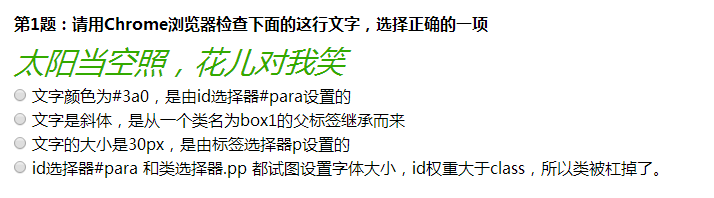
[4.4 行内元素和块级元素可以自由转换](#_Toc13153)

[五、浮动](#_Toc1216)

[5.1 浮动 —— 做并排用的](#_Toc25342)

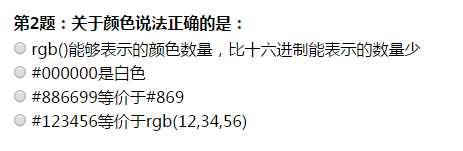
# 一、上节课知识复习

## 第1题 会审查元素



一定要会用Chrome浏览器分析样式，知道每个样式从哪儿来的。

## 第2题 颜色表示



#ff0000这种十六进制，实际上本质也是rgb，只不过用16进制描述0~255，00~ff。rgb中的每一个颜色都和十六进制一一对应的。

#000000是黑色，想象电脑没插电

#aabbcc等价于#abc

#123456等价于rgb(18,52,86);

## 第3题 选择器权重



## 第4题 选择器权重



## 第5题 选择器权重

!important不能影响就近原则，远的标签如果加上!important也干不过近的标签！

!important不能影响继承权重是0，通过继承的标签加上!important也干不过直接选中的标签！



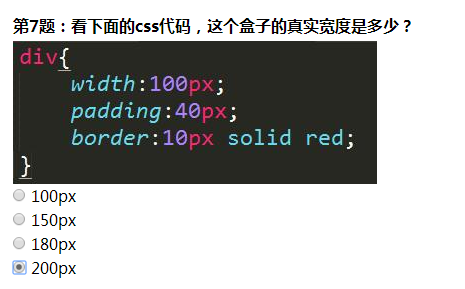
## 第6题 选择器权重

只要开始肉搏了，就是开始比标签权重了，此时!important就生效了



## 第7题 选择器权重

盒模型，一个盒子的真实宽度 = width + 左边padding + 右边padding + 左边border + 右边的border



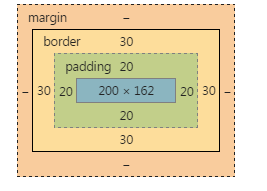
# 二、盒模型

网页中的所有标签都是盒子，都是矩形，都有width、height、padding、border、margin属性的。

最重要的一个知识点，**就是width和真实宽度的关系**。

## 2.1 会看懂盒模型图

|  |
| --- |
| 1. div{ 2. width: 200px; 3. background-color: greenyellow; 4. padding: 20px; 5. border:30px solid red; 6. } |



也就说，padding、border都是外扩的。

## 2.2 padding内边距

padding就是内边距，指的是边框内侧到内容的距离。

padding的设置有很多方法

|  |
| --- |
| 1. padding:50px; |

设置四个方向的padding都是50px;

四个值：

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

等价于单独设置了上、右、下、左的padding分别分10、20、30、40

等价于

|  |
| --- |
| 1. **padding-top**:10px; 2. **padding-right**:20px; 3. **padding-bottom**:30px; 4. **padding-left**:40px; |

三个值：

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

等价于

|  |
| --- |
| 1. padding-top:10px; 2. padding-right:20px; 3. padding-bottom:30px; 4. padding-left:20px; /\*左边的值没有指定，所以和右边相同\*/ |

两个值：

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

等价于

|  |
| --- |
| 1. padding-top:10px; 2. padding-right:20px; 3. padding-bottom:10px; 4. padding-left:20px; |

注意，一些元素默认带有padding：比如ul等等，

为了防止这些默认的padding、margin影响我们制作页面，我们要在页面开头书写：

|  |
| --- |
| 1. <style type="text/css"> 2. **\*{** 3. **padding:0;** 4. **margin:0;** 5. **}** 6. </style> |

就能把所有标签的默认的padding、margin都去掉。

当然这么写有效率问题，别担心，项目课将告诉你最正确的清除默认padding、margin的方法。

## 2.3 border边框

边框有三要素：粗细、线型、颜色。

|  |
| --- |
| 1. border:1px solid red; |

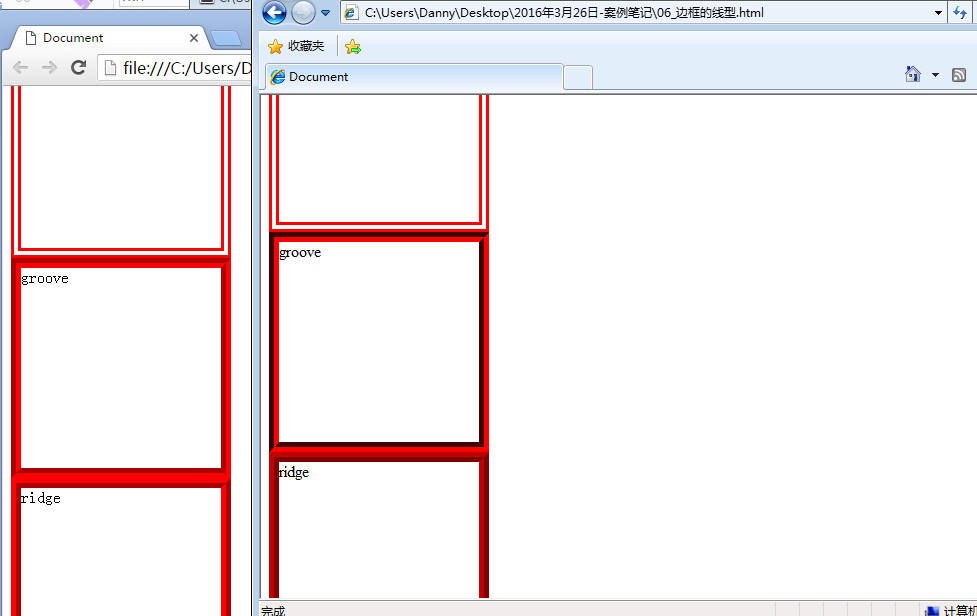
1px 就是边框的宽度， solid就是边框的线型 ， red就是边框的颜色。

线型：



**工作中只许使用solid（实线）和dashed（虚线）**，其他的所有的线型都有兼容性问题，如果想制作其他类型的边框，必须切图！

兼容性问题，IE浏览器中和Chrome浏览器中的边框，明显长得不一样。



注意，三要素分别对应了三个小属性：

|  |
| --- |
| 1. border:10px solid red; |

等价于

|  |
| --- |
| 1. **border-width**:10px; 2. **border-style**:solid; 3. **border-color**:red; |

我们称呼border为“复合属性”。

特别的，**如果我们想为某一个边，单独设置三要素**，那么可以拆分为**12**个小属性：

|  |
| --- |
| 1. border:10px solid red; |

等价于：

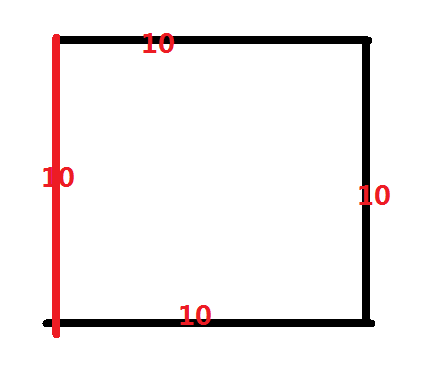
|  |
| --- |
| 1. **border-top-width**:10px; 2. **border-top-style**:solid; 3. **border-top-color**:red; 4. **border-right-width**:10px; 5. **border-right-style**:solid; 6. **border-right-color**:red; 7. **border-bottom-width**:10px; 8. **border-bottom-style**:solid; 9. **border-bottom-color**:red; 10. **border-left-width**:10px; 11. **border-left-style**:solid; 12. **border-left-color**:red; |

更特别的，boder-color是可以按照上右下左的顺序书写：

|  |
| --- |
| 1. border-color:red yellow green; |

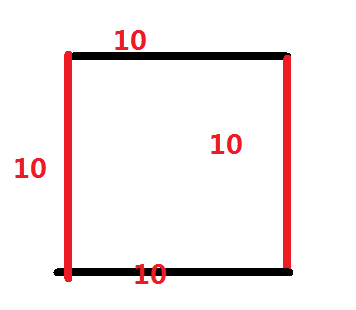
上边是红色， 左边、右边是黄色， 下边是绿色。

小练习1：



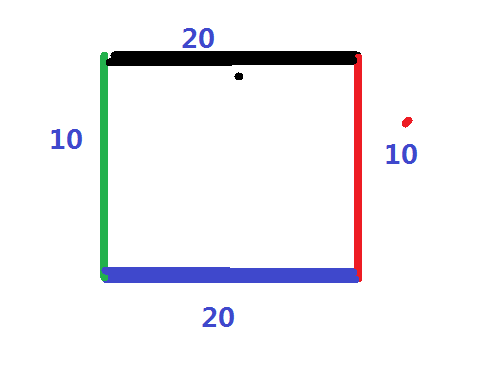
|  |
| --- |
| 1. border:10px solid black; 2. border-left-color:red; |

小练习2：



|  |
| --- |
| 1. border:10px solid black; 2. border-color:black red; |

小练习3：



|  |
| --- |
| 1. border:10px solid red; 2. border-width:20px 10px; 3. border-color:black red blue green; |

如果某一个边框我们不想要，那么就写none

|  |
| --- |
| 1. border-bottom:none; |

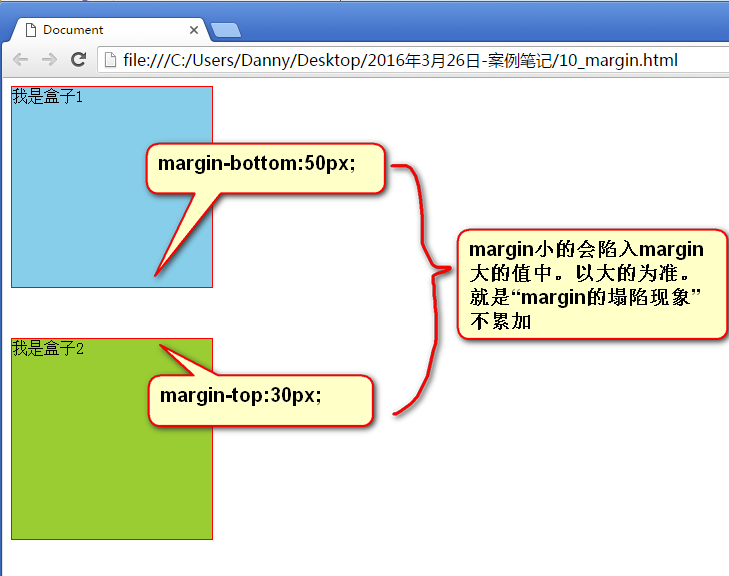
下边框就没有了。

## 2.4 margin外边距

自己这个盒子的边框外侧， 到其他盒子边框外侧的距离

margin有四个方向，分别是 margin-top、margin-right、margin-bottom、margin-left。

塌陷现象：



去掉ul的小圆点，要给ul加

|  |
| --- |
| 1. ul{ 2. list-style:none; 3. } |

# 三、居中策略

## 3.1 文字水平居中

如果想让盒子中的文字水平居中，那么要给盒子设置

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

这个属性继承。

它还有两个可能的值：

|  |
| --- |
| 1. text-align:left; 2. text-align:right; |

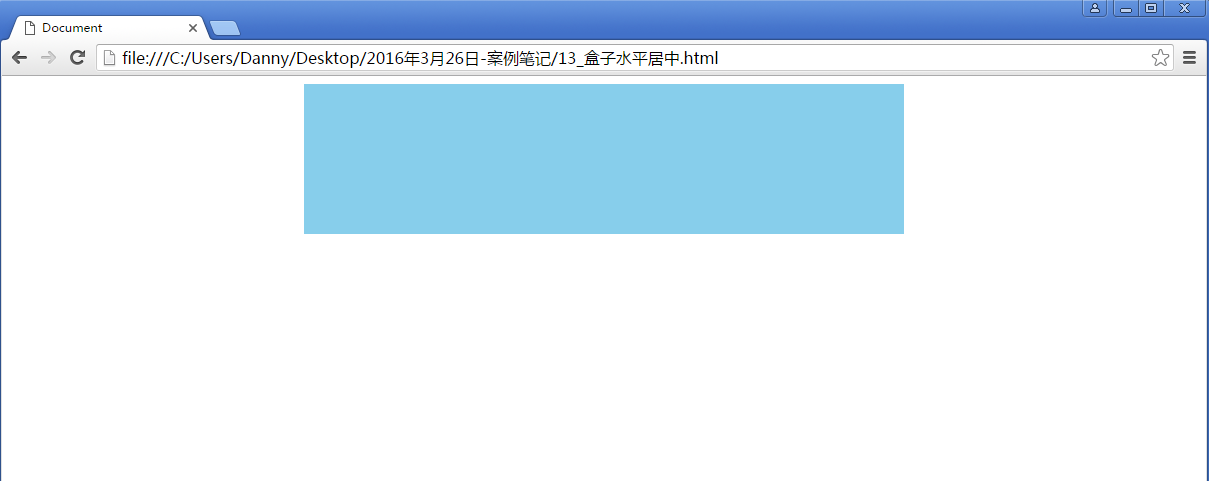
**这个方法，只能居中文本流的东西（文字、图片、表单元素）。一定要记住，这个属性要设置给盒子，不能设置给这些文字、图片、表单元素。**



## 3.2 盒子水平居中

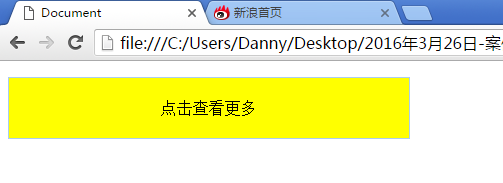
让盒子设置：

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



margin: 0 auto; 是给盒子用的，要加给盒子自己。它将在自己的父元素内部居中。

## 3.3 单行文本的垂直居中



|  |
| --- |
| 1. line-height:盒子的高度; |

行高=盒子高， 此时单行文本就将在盒子中居中。注意，这个方法只能适用于单行文本，多行不适用。

# 四、标准文档流

## 4.1 什么是标准文档流

我们发现，Fireworks作图，可以在任何一个地方落笔；word这个软件则不一样，有一个“光标”位置，你必须在当前光标位置去书写文字。

第二行文字的位置，仰赖于第一行文字，比如第一行文字多，第二行文字就下移了；比如第一行文字大，第二行文字又下移了。

网页也是这样，从左至右、从上到下，有一个“光标”的概念。 标准文档流中，内置了很多性质。

## 4.2 标准文档流有什么性质

### 4.2.1 空白折叠现象

之前讲过，不再赘述。

标签内的文字，不管有多少个空格、换行、tab，都会缩减为同一个空格。

### 4.2.2 高矮不齐，底边对齐



最最重要的性质来了，就是标准文档流中，把元素分为“块级元素”和“行内级元素”两种。

## 4.3 块级元素和行内级元素

我们把HTML标签分为两大类：

● 块级元素（block level）：div、h系列、p、ul、li、dl、dt、dd

● 行内级元素（inline level）：span、a、b、u、i、em、strong

块级元素：

① 能够设置宽度、高度；

② 不能并排；

③ 不设置宽度，那么宽度将默认变为父亲的width

行内级元素：

①不能设置宽度、高度；

②可以并排；

## 4.4 行内元素和块级元素可以自由转换

给任何一个元素，设置

|  |
| --- |
| 1. display:block; |

它将转为块级元素，拥有块级元素的所有性质；

相仿的，给任何一个元素，设置

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

它将转为行内元素，拥有行内元素的所有性质；

有什么用？

把一个块 → 行内 丝毫无用，工作10年都遇不见。

但是行内 → 块 非常有用！

我们发现标准文档流是做不出网页的，因为它太迂腐：能设置宽高的不能并排，能并排的不能设置宽高。

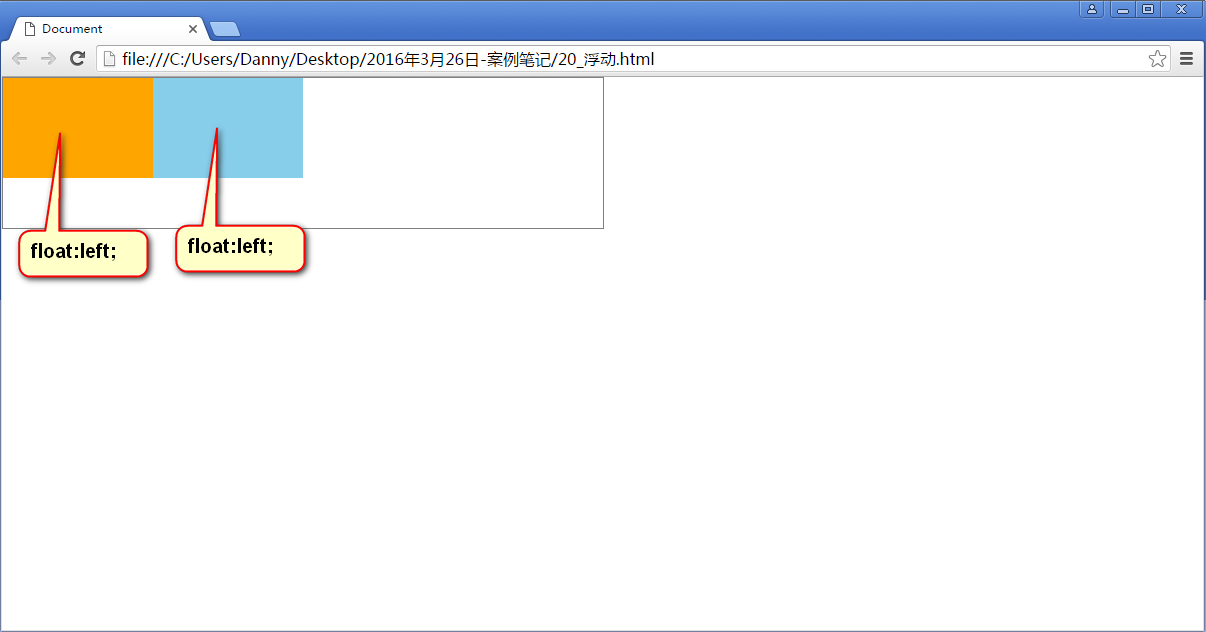
所以脱离标准流！ 脱离标准流一共有三种方法：浮动、绝对定位、相对定位。

# 五、浮动

浮动的理论知识挺多的，头头是道。但是我们为了考虑吸收，我们今天**忽视科学性**，就看一些皮毛的“浮动”。下次课我们深入研究浮动。

今天只用，不研究。

## 5.1 浮动 —— 做并排用的



float就是英语“浮”的意思，

left就是左浮动。

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

right右浮动。

|  |
| --- |
| 1. float:right; |



浮动的元素会去贴父盒子的边，贴边的过程中，如果被它的哥哥们挡住了，就会贴在哥哥的边上。

如果要保证儿子的并排，父亲要有足够的width。

我们下节课将详细研究浮动，记住一个结论：浮动的元素不能撑高父亲了。