CSS 基础入门

第 4 天课堂笔记（本课程共 6 天）

前端与移动开发学院

http://web.itcast.cn

目录

一、复习

盒模型 box model，什么是盒子？所有的标签都是盒子。无论是 div、span、a 都是盒子。图片、表单元素一律看做文本。

盒模型有哪些组成：width、height、padding、border、margin。

width、height 是内容的宽度、高度，想起来丈量包子的比喻、丈量稿纸的比喻。

padding，内边距，边框和文字内容之间的距离。padding 有颜色。表示方法，能够用 padding 综合写，4 个值“上、右、下、左”，3 个值“上、左右、下”，2 个值“上下，左右”。还能按方向拆开，padding-left、padding-top、padding-right、padding-bottom。

border，边框，3 要素，4 条边。3 要素：border-width、border-style、border-color；4 条边：border-top、border-right、border-bottom、border-left。 比如我们要单独设置某一条边，那么就需要写清楚 3 要素：

```
border-top:3px solid red;
```

如果要单独设置要素：

```
border-width:3px;
border-color:red;
border-style:solid;
```

还能拆成最小：

```
border-bottom-style:solid;
```

常用线型：solid、dashed、dotted。

标准文档流：说白了，就是一个“默认”状态。标准文档流中，标签分为两种：块级元素、行内元素。

块级元素：一定是霸占一行的，能设置宽、高，不设置宽度默认就是占满父亲。div、p、h、li

行内元素：和其他行内元素并排，不能设置宽、高，默认宽度就是文字宽度。span、a、b、i、u

能够相互转：

```
display:block;
或者：
display:inline;
```

标准流做不出网页：因为能并排的不能改宽高。所以，要脱离标准流。

浮动：

```
float:left;
或者
float:right;
```

浮动宏观的看，就是做“并排”的。有几个性质：脱标、贴边、字围、收缩。

一个浮动的 a、span ，是不需要设置 display:block; 就能够设置宽高了。因为浮动之后，脱离标准流了，所以标准流里面的法律、规则都不适用了。

二、浮动性质的复习

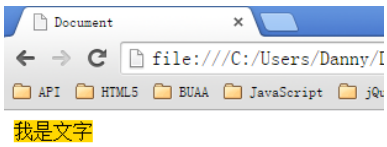
浮动的性质：脱标、贴边、字围、收缩。

收缩：一个浮动的元素，如果没有设置 width，那么将自动收缩为文字的宽度（这点非常像行内元素）。

比如：

```
<style type="text/css">
    div{
        float: left;
        background-color: gold;
    }
</style>
```

这个 div 浮动了，且没有设置宽度，那么将自动缩紧为内容的宽度：



我是文字

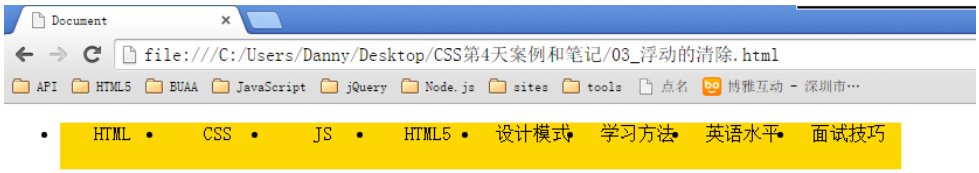
整个网页，就是通过浮动，来实现并排的。

三、浮动的清除

来看一个实验：现在有两个 div，div 身上没有任何属性。每个 div 中都有 li，这些 li 都是浮动的。

```
<div>
    <ul>
        <li>HTML</li>
        <li>CSS</li>
        <li>JS</li>
        <li>HTML5</li>
        <li>设计模式</li>
    </ul>
</div>

<div>
    <ul>
        <li>学习方法</li>
        <li>英语水平</li>
        <li>面试技巧</li>
    </ul>
</div>
```

我们本以为这些 li，会分为两排，但是，第二组中的第 1 个 li，去贴靠第一组中的最后一个 li 了。

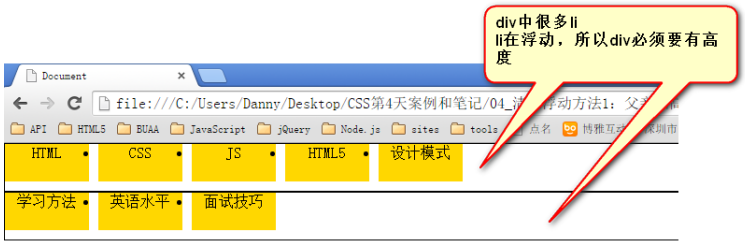


第二个 div 中的 li，去贴第一个 div 中最后一个 li 的边了。

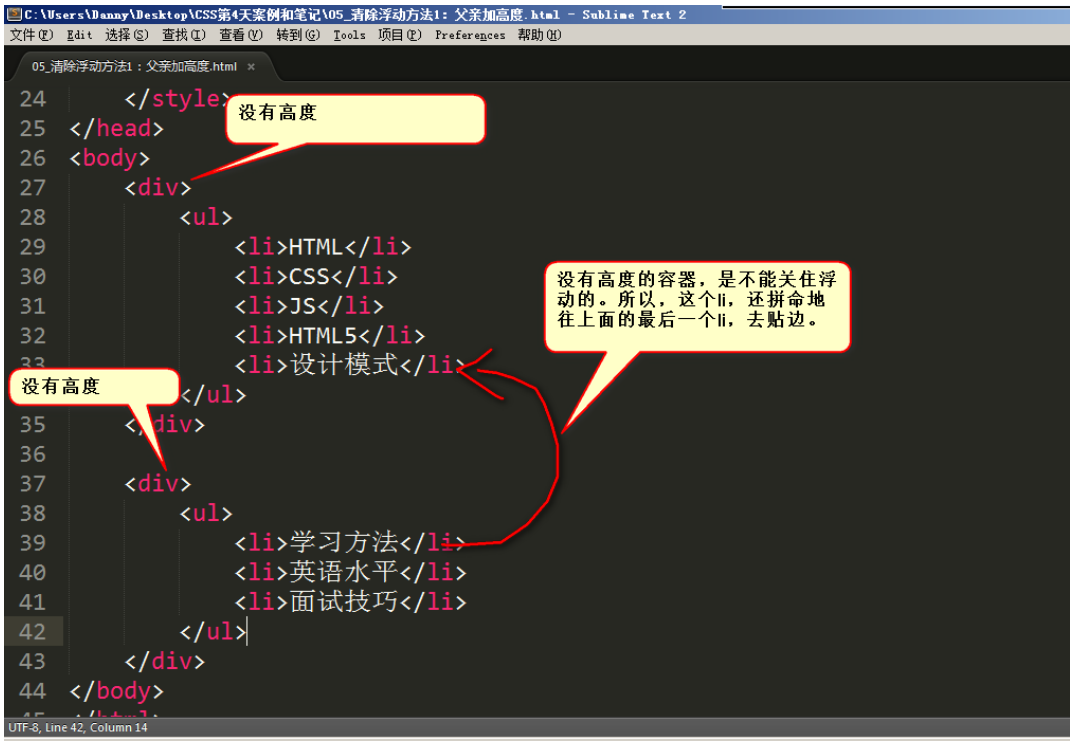
原因就是因为 div 没有高度，不能给自己浮动的孩子们，一个容器。

3.1 清除浮动方法 1：给浮动的元素的祖先元素加高度。

如果一个元素要浮动，那么它的祖先元素一定要有高度。**高度的盒子，才能关住浮动。**



只要浮动在一个有高度的盒子中，那么这个浮动就不会影响后面的浮动元素。所以就是清除浮动带来的影响了。



3.2 清除浮动方法 2：clear:both;

网页制作中，高度 height 很少出现。为什么？因为能被内容撑高！那也就是说，刚才我们讲解的方法 1，工作中用的很少。

脑弄大开：能不能不写 height，也把浮动清除了呢？也让浮动之间，互不影响呢？

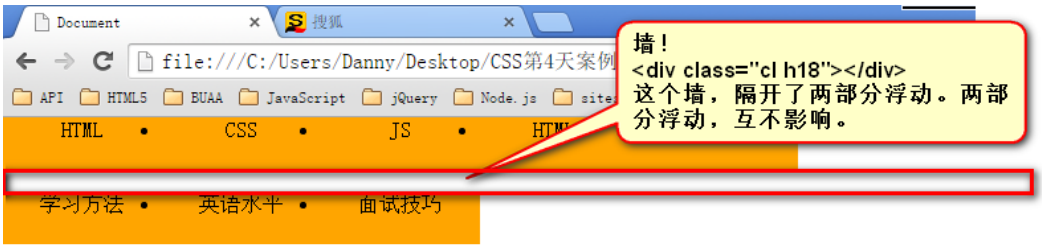
```
    <div>
        <ul>
            <li>HTML</li>
            <li>CSS</li>
            <li>JS</li>
            <li>HTML5</li>
            <li>设计模式</li>
        </ul>
    </div>

    <div class="box2">   → 这个 div 写一个 clear:both;属性
        <ul>
            <li>学习方法</li>
            <li>英语水平</li>
            <li>面试技巧</li>
        </ul>
    </div>
```

```
clear:both;
```

clear 就是清除，both 指的是左浮动、右浮动都要清除。意思就是：清除别人对我的影响。

这种方法有一个非常大的、致命的问题，margin 失效了。

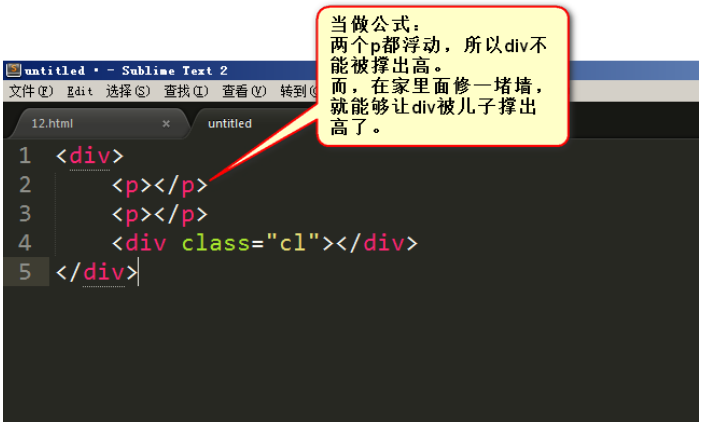
3.3 清除浮动方法 3：隔墙法




```
        <div class="box1">
            <ul>
                <li>HTML</li>
                <li>CSS</li>
                <li>JS</li>
                <li>HTML5</li>
                <li>设计模式</li>
            </ul>
        </div>

        <div class="cl h16"></div>

        <div class="box2">
            <ul>
                <li>学习方法</li>
                <li>英语水平</li>
                <li>面试技巧</li>
            </ul>
        </div>
```

```
.cl{
    clear: both;
}
.h16{
    height: 16px;
}
```

近些年，有演化出了“内墙法”：

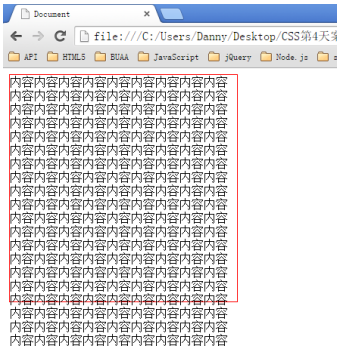
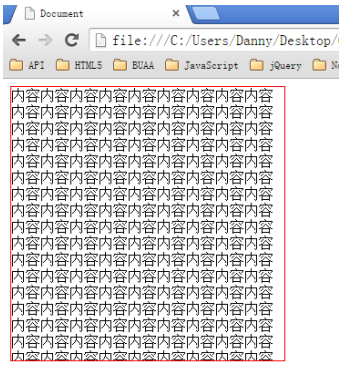


3.4 清除浮动方法 4：overflow:hidden;

overflow 就是“溢出”的意思， hidden 就是“隐藏”的意思。

```
overflow:hidden;
```

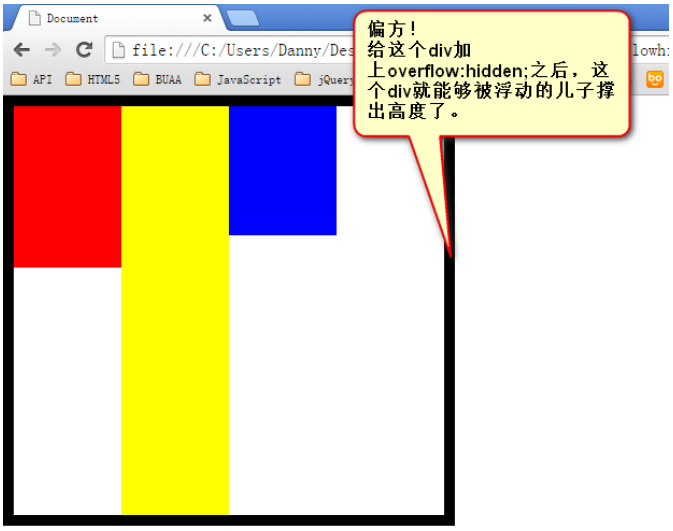
表示“溢出隐藏”。所有溢出边框的内容，都要隐藏掉。

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。但是，前端开发工程师又发现了，它能做偏方。

一个父亲不能被自己浮动的儿子，撑出高度。但是，只要给父亲加上 overflow:hidden; 那么，父亲就能被儿子撑出高了。这是一个偏方。

```
div{
    width: 400px;
    border: 10px solid black;
    overflow: hidden;
}
```

3.5 清除浮动总结与案例

总结一下：

1） 加高法：

浮动的元素，只能被有高度的盒子关住。 也就是说，如果盒子内部有浮动，这个盒子有高，那么妥妥的，浮动不会互相影响。但是，工作上，我们绝对不会给所有的盒子加高度，这是因为麻烦，并且不能适应页面的快速变化。

```
<div>     → 设置 height
    <p></p>
    <p></p>
    <p></p>
</div>

<div>     → 设置 height
    <p></p>
    <p></p>
    <p></p>
</div>
```

2) clear:both;法

最简单的清除浮动的方法，就是给盒子增加 clear:both；表示自己的内部元素，不受其他盒子的影响。

```
<div>
    <p></p>
    <p></p>
    <p></p>
</div>

<div>     → clear:both;
    <p></p>
    <p></p>
    <p></p>
</div>
```

浮动确实被清除了，不会互相影响了。但是有一个问题，就是 margin 失效。两个 div 之间，没有任何的间隙了。

3）隔墙法：

在两部分浮动元素中间，建一个墙。隔开两部分浮动，让后面的浮动元素，不去追前面的浮动元素。

墙用自己的身体当做了间隙。

```
<div>
    <p></p>
    <p></p>
    <p></p>
</div>

<div class="cl h10"></div>

<div>
    <p></p>
    <p></p>
    <p></p>
</div>
```

我们发现，隔墙法好用，但是第一个 div，还是没有高度。如果我们现在想让第一个 div，自动的根据自己的儿子，撑出高度，我们就要想一些“小伎俩”，“奇淫技巧”。

内墙法：

```
<div>
    <p></p>
    <p></p>
    <p></p>
    <div class="cl h10"></div>
</div>

<div>
    <p></p>
    <p></p>
    <p></p>
</div>
```

内墙法的优点就是，不仅仅能够让后部分的 p 不去追前部分的 p 了，并且能把第一个 div 撑出高度。这样，这个 div 的背景、边框就能够根据 p 的高度来撑开了。

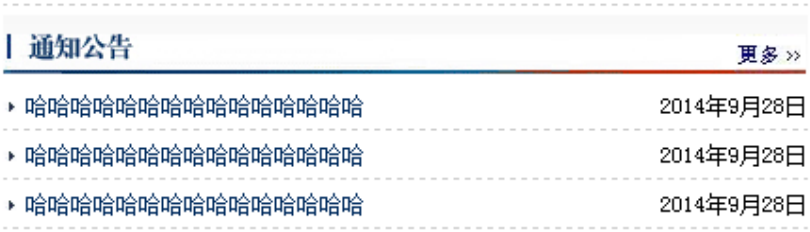
4）overflow:hidden;

这个属性的本意，就是将所有溢出盒子的内容，隐藏掉。但是，我们发现这个东西能够用于浮动的清除。

我们知道，一个父亲，不能被自己浮动的儿子撑出高度，但是，如果这个父亲加上了 overflow:hidden；那么这个父亲就能够被浮动的儿子撑出高度了。这个现象，不能解释，就是浏览器的小偏方。

并且,overflow:hidden;能够让 margin 生效。

清除浮动的案例：



实践中，遇见了清除浮动的问题：

3.6 浏览器兼容问题

上述知识点遇见的浏览器兼容问题

第一，IE6，不支持小于 12px 的盒子，任何小于 12px 的盒子，在 IE6 中看都大

解决办法很简单，就是将盒子的字号，设置小（小于盒子的高），比如 0px。

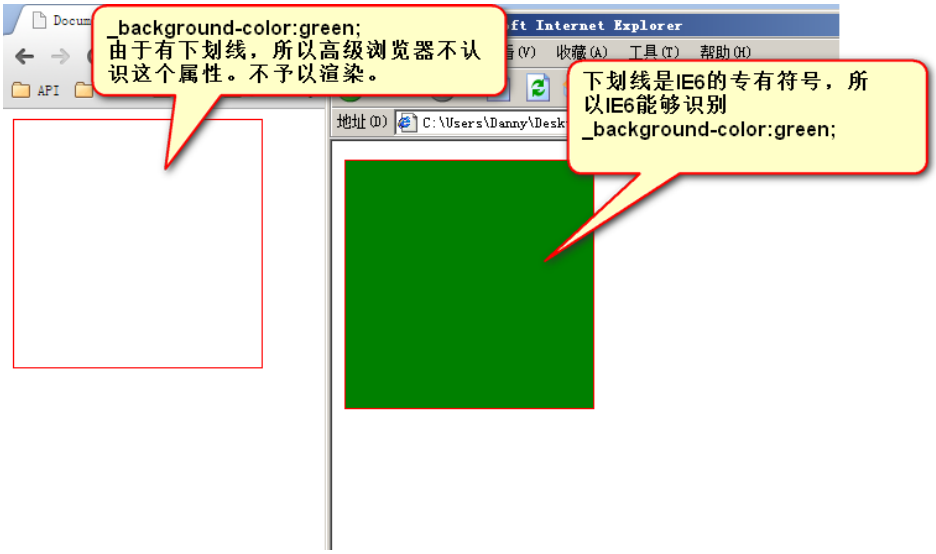
```
height: 4px;
_font-size: 0px;
```

我们现在介绍一下浏览器 hack。hack 就是“黑客”，就是使用浏览器提供的后门，针对某一种浏览器做兼容。

IE6 留了一个后门，就是只要给 css 属性之前，加上下划线，这个属性就是 IE6 认识的专有属性。

比如：

```
_background-color: green;
```



解决微型盒子，正确写法：

```
height: 10px;
_font-size:0;
```

第二，IE6 不支持用 overflow:hidden;来清除浮动的

解决办法，以毒攻毒。追加一条

```
_zoom:1;
```

完整写法：

```
overflow: hidden;
_zoom:1;
```

实际上，_zoom:1;能够触发浏览器 hasLayout 机制。这个机制，不要深究了，因为就 IE6 有。我们只需要让 IE6 好用，具体的实现机制，有兴趣的同学，自行百度。

强调一点， overflow:hidden;的本意，就是溢出盒子的 border 的东西隐藏，这个功能是 IE6 兼容的。不兼容的是 overflow:hidden;清除浮动的时候。

我们刚才学习了两个 IE6 的兼容问题，这两个 IE6 的兼容问题，都是通过多写一条 hack 来解决的。

这个我们称为伴生属性。

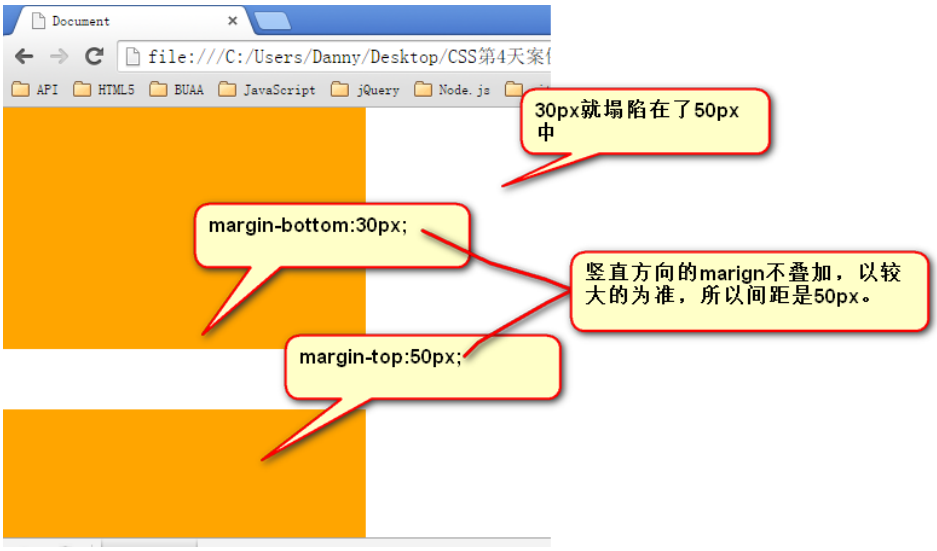
```
height:6px;
_font-size:0;
```

```
overflow:hidden;
_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：

```
1    margin-left: auto;
2    margin-right: auto;
```

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

```
text-align:center;
```

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

普及一下知识，text-align 还有

```
text-align:left;      没啥用，因为默认居左
text-align:right;     文本居右
```

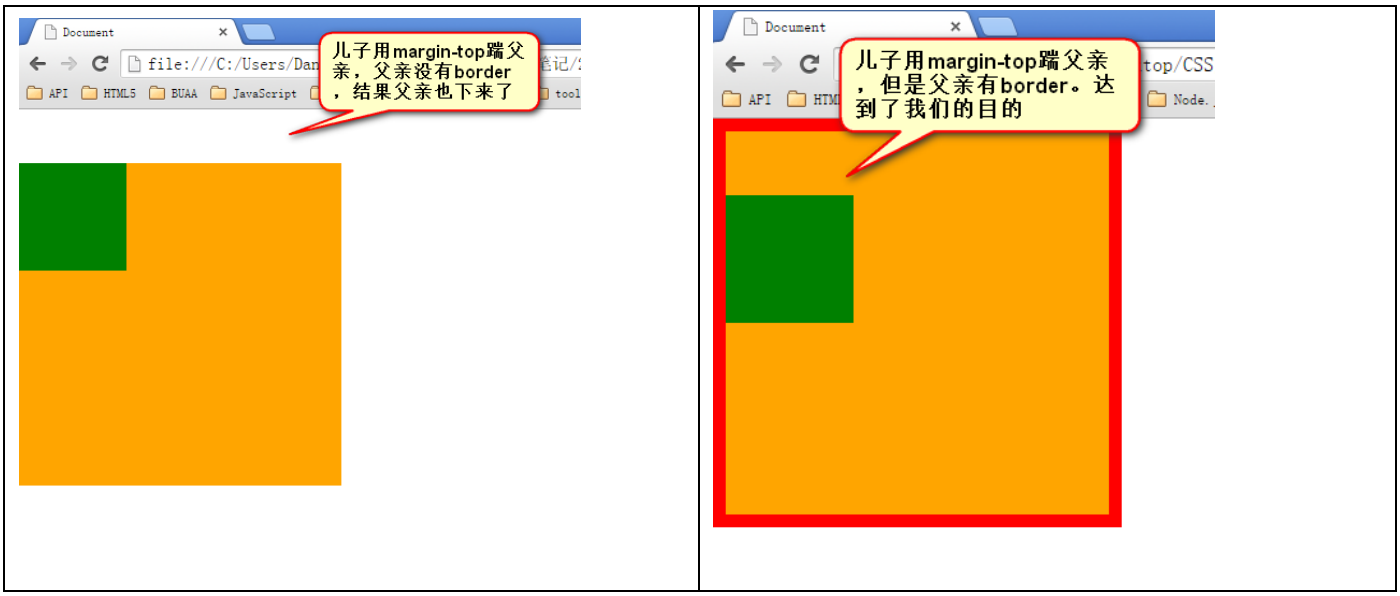
4.3 善于使用父亲的 padding，而不是儿子的 margin

如果父亲没有 border，那么儿子的 margin 实际上踹的是“流”，踹的是这“行”。所以，父亲整体也掉下来了

这个 p 有一个 margin-top 踹父亲，试图将自己下移

```
<div>
    <p></p>
</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； 最好不要用这个 marign 表达父子之间的距离。

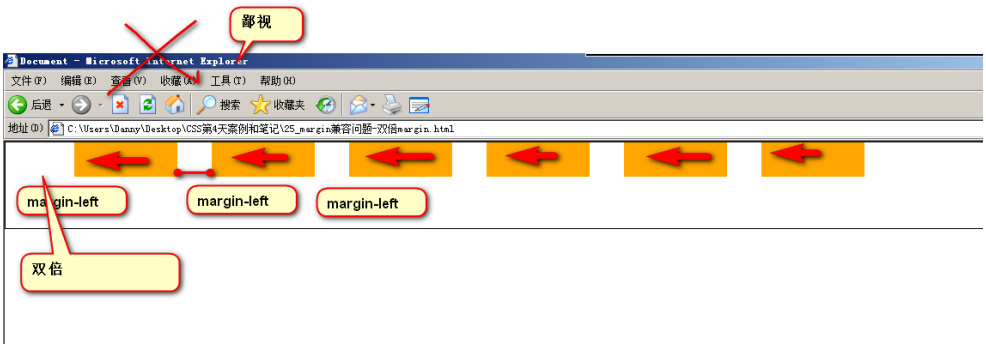
所以，我们一定要善于使用父亲的 padding，而不是儿子的 margin。

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。

所以，你就会发现，我们特别喜欢，浮动的方向和 margin 的方向相反。并且，前端开发工程师，把这个当做习惯了。

```
1                float: left;
2                margin-right: 40px;
```

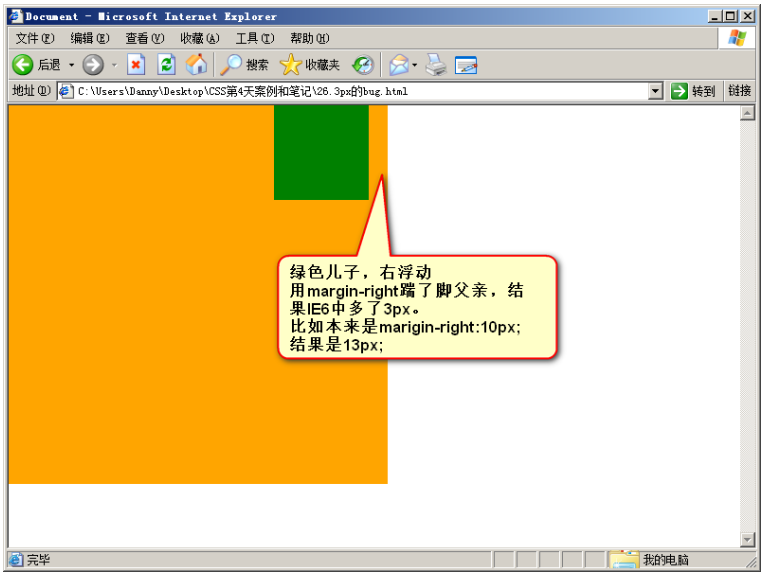
2）使用 hack（没必要，别惯着这个 IE6）

单独给队首的元素，写一个一半的 margin

```
1    <li class="no1"></li>
```

```
1    ul li.no1{
2        _margin-left:20px;
3    }
```

IE6 的 3px bug



解决办法：

不用管，因为根本就不允许用儿子踹父亲。所以，如果你出现了 3px bug，说明你的代码不标准。

IE6，千万不要跟他死坑、较劲，它不配。 格调要高，老师给你讲，就是为了增加面试的成功率。不是为了让你成为 IE6 的专家。

五、Fireworks 和精确盒子还原

fireworks 是 Adobe 公司的一个设计软件。功能非常多，我们今天用啥讲啥。

Fireworks 的默认文件格式是 png。

新建 ctrl+N。

分辨率是 72 像素/英寸 （咱们不用知道，因为设计师把设计图给你）

标尺的快捷键，是 ctrl+alt+r。

css 中，任何文本都有行高。行高就是

```
1    line-height
```

属性。顾名思义，就是行的高度。

首行空两个汉字的格，单位比较奇怪，叫做 em，em 就是汉字的一个宽度。

```
1    text-indent:2em;
```

indent 就是“缩进”的意思。