

DIV+CSS快速入门



主讲：刘金榜

智搏网联学习群：114808677

网络教程：www.w3school.com.cn

课件下载地址：www.hifoge.com/css

第一章 CSS简介



- **CSS 概述**
- CSS 指层叠样式表 (Cascading Style Sheets)
- 样式定义如何显示 HTML 元素
- 样式通常存储在样式表中
- 外部样式表可以极大提高工作效率

万维网联盟 (W3C)，非营利的标准化联盟

1996年12月，发布了CSS 1.0规范。

1998年 5月，发布了CSS 2.0规范。

目前有两个新版本正在处于工作状态，即CSS 2.1版和3.0版

第一章 CSS简介



W3C并没有强制软件厂商的产品必须符合规范，因此目前流行的浏览器都没有完全符合规范，这就给设计师设计网页带来了一些难题。

目前最主流的浏览器是：

IE6.0、IE7.0、IE8.0、Firefox

市场占有率为：99%

以这4种浏览器为目标，已经完全可以做出显示非常一致的CSS布局页面。随着技术的发展，各种浏览器都会逐渐在这方面做更多的努力，相信情况会越来越好。

第一章 CSS简介



- 什么是DIV+CSS?

DIV元素是用来为HTML文档内大块（block-level）的内容提供结构和背景的元素。

CSS是Cascading Style Sheets(层叠样式表)的简称。

- 通俗来说:

如果把一个网页比作一栋楼房，div就是楼房的框架结构，一个个的砖头就是一个一个的标签元素（如图片，文字），CSS就是水泥和玻璃，把砖瓦合理的跟框架连接在一起。

DIV+CSS 其实就是布局的一个方法，就是像盖房子的方法一样。

第一章 CSS简介



- CSS的优势:

- 1、缩减代码，提高页面浏览速度。
- 2、结构清晰，对搜索引擎更加友好。
- 3、支持各种浏览器，兼容性好。
- 4、简单的修改，缩短改版时间。
- 5、强大的字体控制和排版能力。
- 6、使用CSS可以结构化HTML，提高易用性。

.....

各种好处！

第一章 CSS简介



- 学习CSS，需要什么基础
 - 1) 对HTML和网页设计有初步的基础
 - 2) 具有钻研的**精神**和**热情**

其中第1点的权重占**10%**

第2点的权重点**90%**

第二章 CSS基础语法



- 语法是CSS学习的开始，不要小看它。
- 因为**成也语法，败也语法！**
- CSS 语法由三部分构成：
选择器、**属性**、**值**
例如：`p {color: red;}`
`h1 { border:1px solid red;}`

第二章 CSS基础语法



- **选择器**：通常是你希望定义的 HTML 元素或标签，
 - **属性**：是你希望改变的属性，并且每个属性都有一个值。
 - 属性和值被冒号分开，并由花括号包围，这样就组成了一个完整的样式声明。
-
- `body {color: blue; }`
 - 上面这行代码的作用是将 `body` 元素内的文字颜色定义为蓝色。在上述例子中，**`body` 是选择器**，而包括在花括号内的部分是声明。声明依次由两部分构成：属性和值，**`color` 为属性，`blue` 为值。**

第二章 CSS基础语法



- 选择器的分类
 - 标记选择器
 - ID选择器
 - 类别选择器
 - 复合选择器
 - 交集选择器
 - 并集选择器
 - 后代选择器

第二章 CSS基础语法



- 选择器——标记选择器

Html结构中：<p>标记选择器</p>

CSS中：

```
p { text-align: center;  
    color: black;  
}
```

此例中CSS中的p标记即为标记选择器。

第二章 CSS基础语法



- 选择器——ID选择器

Html结构中：<div id= “box” >这是ID选择器</div>

CSS中：

```
#box { text-align: center;
        color: black;
    }
```

此例中CSS中的#box即为ID选择器。

注意：ID选择器用#

第二章 CSS基础语法



- 选择器——类别选择器

Html结构中：<div class= “box” >这类别选择器</div>

CSS中：

```
.box { text-align: center;
        color: black;
    }
```

此例中CSS中的.box即为类别选择器。

注意：类别选择器用“ . ”

第二章 CSS基础语法



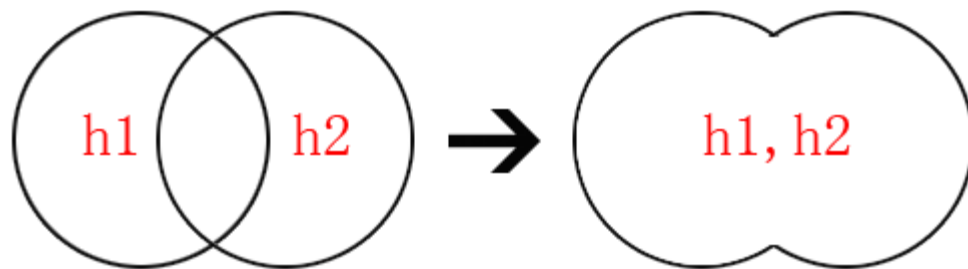
- 选择器——复合选择器__并集选择器

Html结构中：<h1>这将是并集选择器</h1>

<h2>这也将是并集选择器</h2>

CSS中：

```
h1,h2{ text-align: center; background-color:orange; }
```



此例中CSS中的“h1,h2”即为复合选择器中的**并集选择器**
并集选择器是多个选择器通过逗号连接而成的，并集选择器可以同时声明风格相同的样式。

第二章 CSS基础语法



- 选择器——复合选择器_交集选择器

Html结构中：<p id= “box” >这是交集选择器</p>

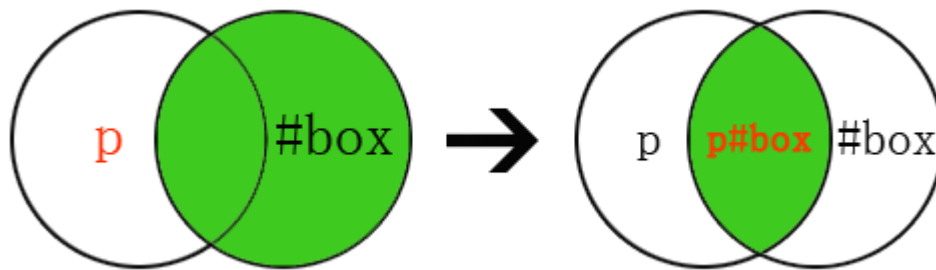
<p class= “box” >这也是交集选择器</p>

CSS中：

```
p{ background-color:orange;}
```

```
p#box{ color : red; }
```

```
p.box{ color:blue;}
```



此例中CSS中的“p#box”和“p.box”都是复合选择器中的交集选择器。第一个必须是标记选择器，第二个必须是类别选择器或ID选择器，这两个选择器之间不能有空格，必须连续书写。

第二章 CSS基础语法



- 选择器——后代选择器

Html结构中： `<div id=“father” >这是父元素`
`<p>这是子元素</p>`
`</div>`

CSS中： `#father{ height:150px; background:orange;}`
`#father p{ width:50px;`
`height:50px;`
`background:bule;}`

此例中CSS中的“`#father p`”为后代选择器。

注意：后代选择器的写法：外层的标记写在前面，内层的标记写在后面，之间用空格分隔。此p标记就是#father的后代。

第二章 CSS基础语法



- CSS继承

CSS继承指的是子标记会继承父标记的所有样式风格，并可以在父标记样式风格的基础上再加以修改，产生新的样式，而子标记的样式风格完全不会影响父标记。

```
<div class="father">
```

这是父元素的红色

```
<div class="son">子元素继承了父元素的红色</div>
```

```
</div>
```

若对.son设置为紫色，.son{color:purple;}，则并不会影响父元素的颜色。

第二章 CSS基础语法



- CSS的层叠特性

CSS的全名：层叠样式表

“层叠”是什么意思？为何这个词如此重要，以至于要出现在它的名称里？

在一个结构复杂的页面中，某一个元素有可能会从很多地方获得样式，就完全有可能使代码变得非常混乱。可能出现在“无法找到某一个元素的样式来自于哪条规则”的情况。因此，必须要充分理解CSS中的“层叠”的原理。

- 那么我们如何解决层叠样式中的“冲突”问题呢？

第二章 CSS基础语法



- CSS的层叠特性（续）
- 优先级：
行内样式 > ID样式 > 类别样式 > 标记样式

注意：

千万不要混淆了“层叠”与“继承”，二者完全不同！

第二章 CSS基础语法



- 如何创建CSS ?
- 插入样式表的方法有三种：
- 方法一：外部样式表
- 当样式需要应用于很多页面时，外部样式表将是理想的选择。在使用外部样式表的情况下，你可以通过改变一个文件来改变整个站点的外观。每个页面使用 <link> 标签链接到样式表。<link> 标签在（文档的）头部：

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="mystyle.css" />
```

```
</head>
```

外部样式表可以在任何文本编辑器进行编辑。文件不能包含任何的 html 标签。样式表应该以 .css 扩展名进行保存。

第二章 CSS基础语法



- 创建CSS方法二： **内部样式表**

当单个文档需要特殊的样式时，就应该使用内部样式表。你可以使用 `<style>` 标签在头部分定义内部样式表，就像这样：

```
<head>
```

```
<style type="text/css">
```

```
hr { color:red;}
```

```
p{ margin-left: 20px;}
```

```
body{background-color:blue;} </style>
```

```
</head>
```

第二章 CSS基础语法



- 创建CSS方法三：行内样式

- 由于要将表现和内容混杂在一起，内联样式会损失掉样式表的许多优势。请慎用这种方法，例如当样式仅需要在一个元素上应用一次时。
- 要使用内联样式，你需要在相关的标签内使用样式（style）属性。Style 属性可以包含任何 CSS 属性。本例展示如何改变段落的颜色和左外边距：

```
<p style="color:red; margin-left: 20px">
```

这是行内样式，也称为内联样式

```
</p>
```

第二章 CSS基础语法



- 多重样式如何显示出来？
- 例如，**外部样式表**拥有针对 h3 选择器的三个属性：

```
h3{ color:red;
      text-align:left;
      font-size:8pt;
}
```

- 而**内部样式表**拥有针对 h3 选择器的两个属性：

```
h3{
      text-align: right;
      font-size: 20pt;
}
```

假如拥有内部样式表的这个页面同时与外部样式表链接，那么 **h3 得到的样式**是：

```
h3{
      color: red;
      text-align: right;
      font-size: 20pt;
}
```

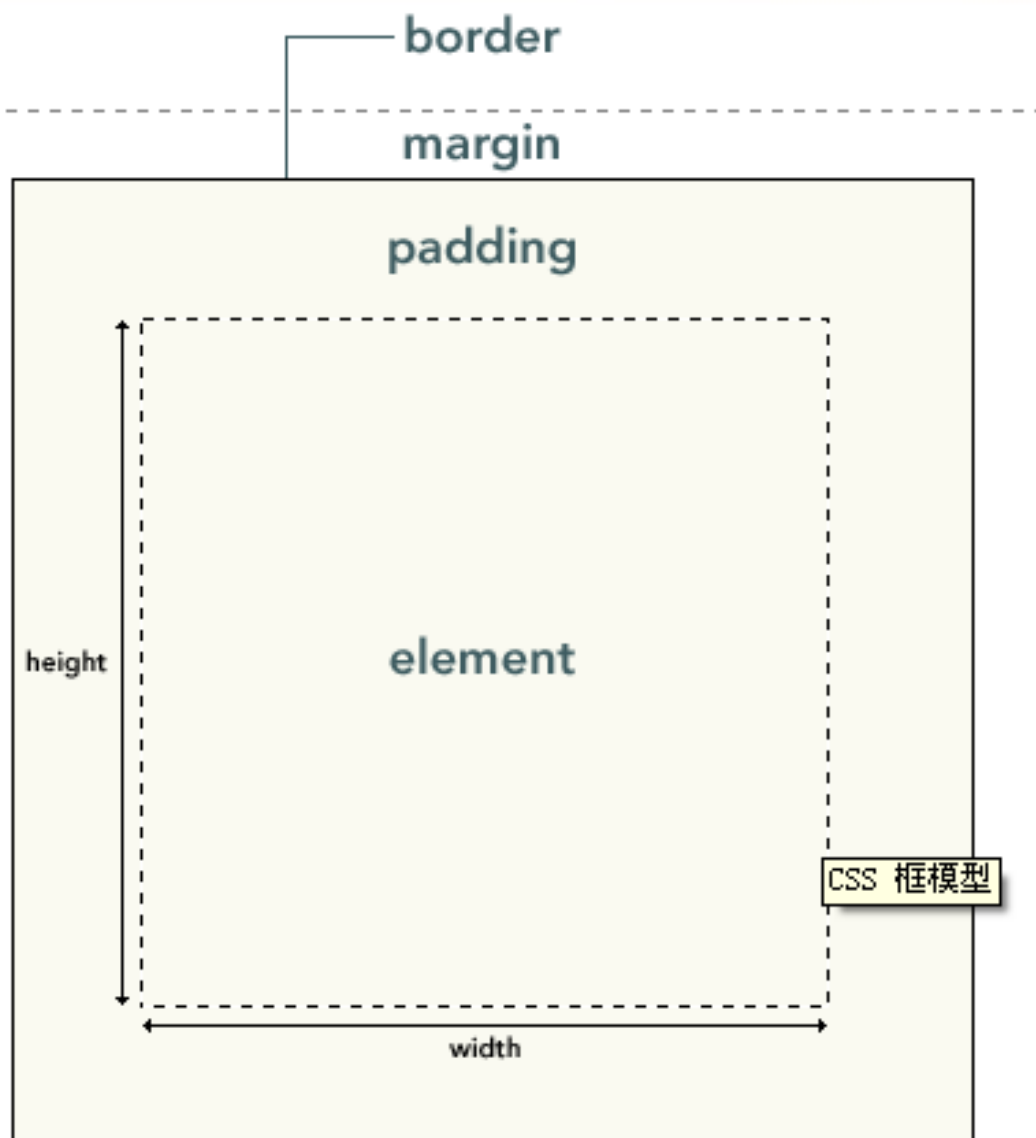
即颜色属性将被继承于外部样式表，而文字排列（text-align）和字体尺寸（font-size）会被内部样式表中的规则取代。

第三章 盒子模型



- 盒子模型是CSS控制页面时一个很重要的概念。只有很好地掌握了盒子模型以及其中每个元素的用法，才能真正地控制好页面中的各个元素。本章主要介绍盒子模型的基本概念，并讲解CSS定位的基本方法。
- 所有页面中的元素都可以看成是一个盒子，占据着一定的页面空间。
- 一个页面由很多这样的盒子组成，这些盒子之间会互相影响，因此掌握盒子模型需要从两方面来理解。一是理解一个孤立盒子的内部结构，二是理解多个盒子之间的相互关系。

第三章 盒子模型



- CSS 框模型：
包括：

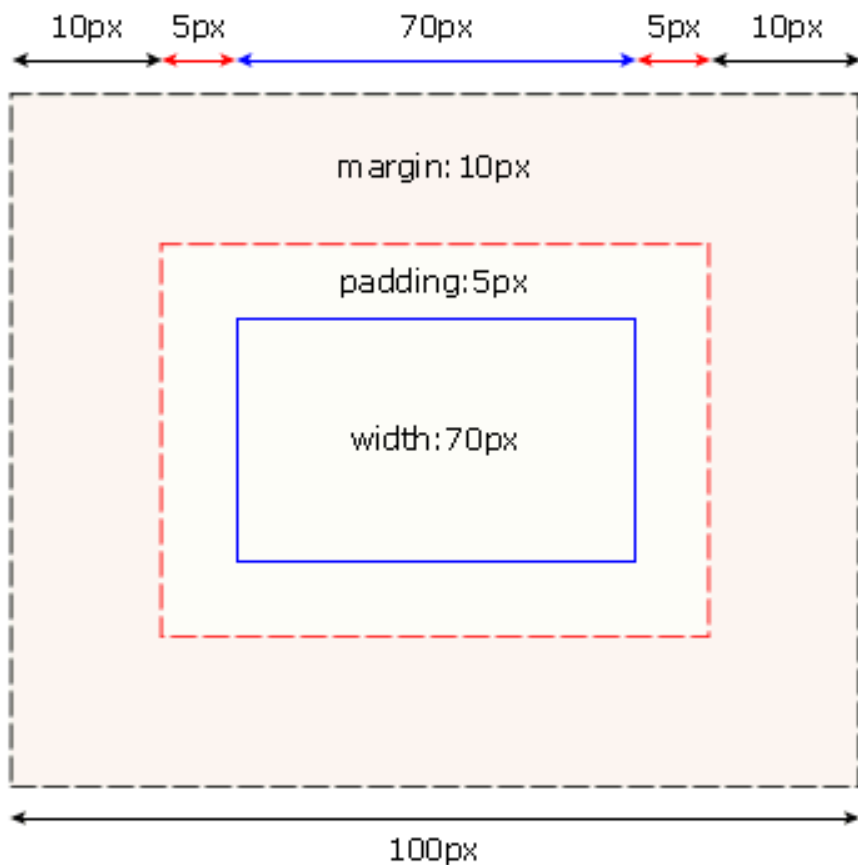
边框

内边距

外边距

元素内容

第三章 盒子模型



在 CSS 中，width 和 height 指的是内容区域的宽度和高度。增加内边距、边框和外边距不会影响内容区域的尺寸，但是会增加元素框的尺寸。

假设框的每个边上有 10 个像素的外边距和 5 个像素的内边距。如果希望这个元素框达到 100 个像素，就需要将内容的宽度设置为 70 像素，如图。



第三章 盒子模型__边框属性（1）



属性	描述
border	简写属性。作用是在一个声明中用来设置四个边框的所有属性。
<i>border-style</i>	设置四个边框的样式，可以设置一到四个样式。
<i>border-width</i>	设置四个边框的宽度，可以设置一到四个值。
<i>border-color</i>	设置四个边框的颜色，可以设置一到四个颜色。
border-bottom	简写属性。作用是在一个声明中用来设置下边框的所有属性。
<i>border-bottom-color</i>	设置元素的下边框的颜色。
<i>border-bottom-style</i>	设置元素的下边框的样式。
<i>border-bottom-width</i>	设置元素的下边框的宽度。

温馨提示：推荐使用简写属性，灰色属性不推荐使用。

第三章 盒子模型__边框属性（2）



属性	描述
<code>border-left</code>	简写属性。用于在一个声明中设置左边框的所有属性。
<code>border-left-color</code>	设置元素的左边框的颜色。
<code>border-left-style</code>	设置元素的左边框的样式。
<code>border-left-width</code>	设置元素的左边框的宽度。
<code>border-right</code>	简写属性。将所有用于右边框的属性设置于一个声明中。
<code>border-right-color</code>	设置元素的右边框的颜色。
<code>border-right-style</code>	设置元素的右边框的样式。
<code>border-right-width</code>	设置元素的右边框的宽度。

温馨提示：推荐使用简写属性，灰色属性不推荐使用。

第三章 盒子模型__边框属性（3）



属性	描述
<code>border-top</code>	简写属性。将所有用于上边框的属性设置于一个声明中。
<i><code>border-top-color</code></i>	设置元素的上边框的颜色。
<i><code>border-top-style</code></i>	设置元素的上边框的样式。
<i><code>border-top-width</code></i>	设置元素的上边框的宽度。

温馨提示：推荐使用简写属性，灰色属性不推荐使用。

第三章 盒子模型__边框(border-style)



border-style	描述
none	定义无边框。
dotted	定义点状边框。在大多数浏览器中呈现为实线。
dashed	定义虚线。在大多数浏览器中呈现为实线。
solid	定义实线。
double	定义双线。双线的宽度等于 border-width 的值。
groove	定义 3D 凹槽边框。其效果取决于 border-color 的值。
ridge	定义 3D 垄状边框。其效果取决于 border-color 的值。
inset	定义 3D inset 边框。其效果取决于 border-color 的值。
outset	定义 3D outset 边框。其效果取决于 border-color 的值。

注意：IE浏览器不支持的**border-style**效果，在实际制作网页的时候，不推荐使用。

第三章 盒子模型__边框(border-width)



border-width	描述
thin	定义细的边框。
medium	默认。定义中等的边框。
thick	定义粗的边框。
length	允许您自定义边框的宽度。
inherit	规定应该从父元素继承边框宽度。

温馨提示：灰色属性不推荐使用。

第三章 盒子模型__边框(border-color)



border-color	描述
color_name	规定颜色值为颜色名称的边框颜色（比如 red）。
hex_number	规定颜色值为十六进制值的边框颜色（比如 #ff0000）。
<i>rgb_number</i>	规定颜色值为 rgb 代码的边框颜色（比如 rgb(255, 0, 0)）。
transparent	默认值。边框颜色为透明。
inherit	规定应该从父元素继承边框颜色。

温馨提示：灰色属性不推荐使用。

第三章 盒子模型__内边距 (padding)



- 元素的内边距在边框和内容区之间。
- CSS padding 属性定义元素边框与元素内容之间的空白区域。

属性	描述
padding	简写属性。作用是在一个声明中设置元素的所内边距属性。
padding-bottom	设置元素的下内边距。
padding-left	设置元素的左内边距。
padding-right	设置元素的右内边距。
padding-top	设置元素的上内边距。

第三章 盒子模型__内边距 (padding)

内边距padding指文本边框与文本之间的距离，四个值沿顺时针方向，顺序依次是上、右、下、左



例子 1

```
padding:10px 5px 15px 20px;
```

- 上内边距是 10px
- 右内边距是 5px
- 下内边距是 15px
- 左内边距是 20px

例子 2

```
padding:10px 5px 15px;
```

- 上内边距是 10px
- 右内边距和左内边距是 5px
- 下内边距是 15px

例子 3

```
padding:10px 5px;
```

- 上内边距和下内边距是 10px
- 右内边距和左内边距是 5px

例子 4

```
padding:10px;
```

- 所有 4 个内边距都是 10px

第三章 盒子模型__外边距 (margin)



围绕在元素边框的空白区域是外边距。设置外边距会在元素外创建额外的“空白”。

设置外边距的最简单的方法就是使用 `margin` 属性，这个属性接受任何长度单位、百分数值甚至负值。

属性	描述
<code>margin</code>	简写属性。在一个声明中设置所有外边距属性。
<code>margin-bottom</code>	设置元素的下外边距。
<code>margin-left</code>	设置元素的左外边距。
<code>margin-right</code>	设置元素的右外边距。
<code>margin-top</code>	设置元素的上外边距。

第三章 盒子模型__外边距 (margin)

外边距Margin指的是元素与元素之间的距离，四个值沿顺时针方向，顺序依次是上、右、下、左



例子 1

```
margin:10px 5px 15px 20px;
```

- 上外边距是 10px
- 右外边距是 5px
- 下外边距是 15px
- 左外边距是 20px

例子 2

```
margin:10px 5px 15px;
```

- 上外边距是 10px
- 右外边距和左外边距是 5px
- 下外边距是 15px

例子 3

```
margin:10px 5px;
```

- 上外边距和下外边距是 10px
- 右外边距和左外边距是 5px

例子 4

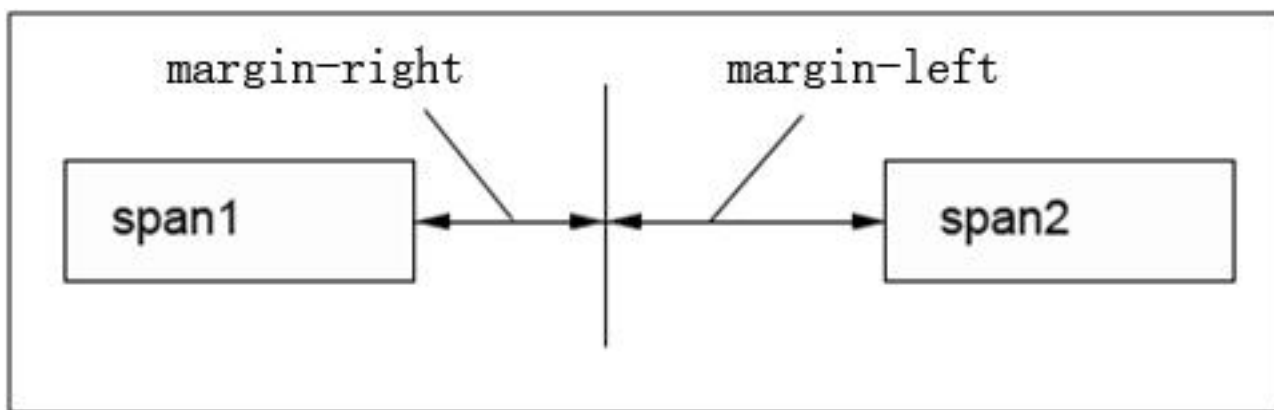
```
margin:10px;
```

- 所有 4 个外边距都是 10px

第三章 盒子模型_盒子间的关系



- 行内元素之间的水平距离

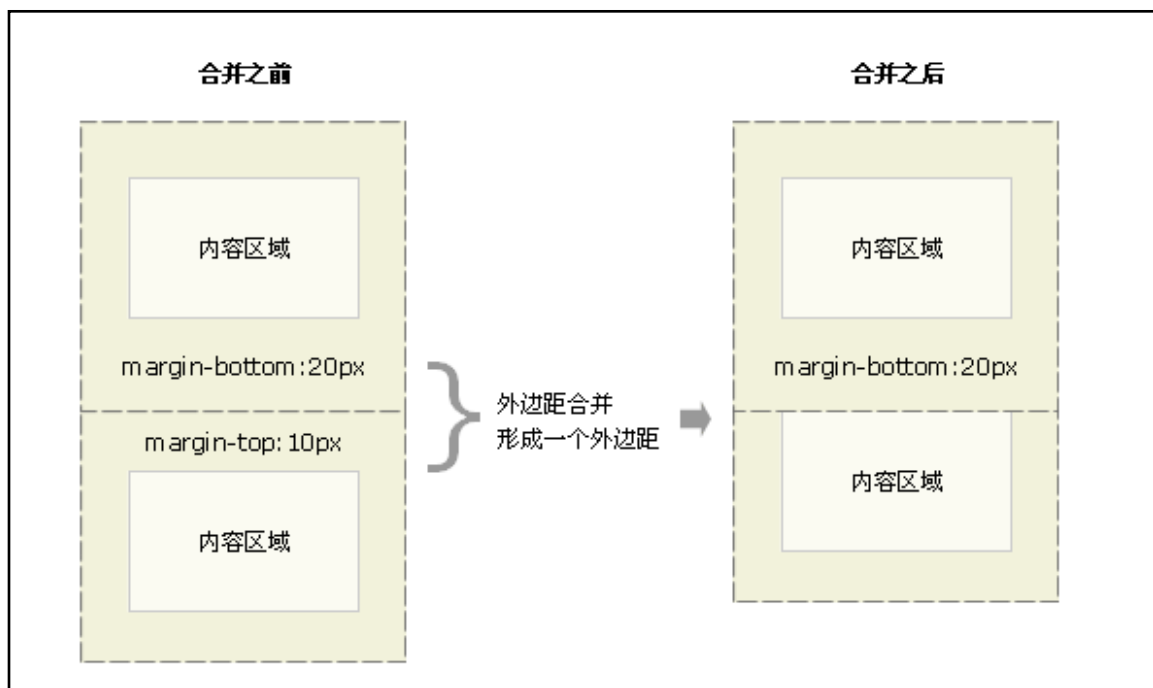


当两个行内元素并排紧邻时，它们之间的距离为margin-left和margin-right之和。

第三章 盒子模型__盒子间的关系



- 块级元素之间的竖直距离

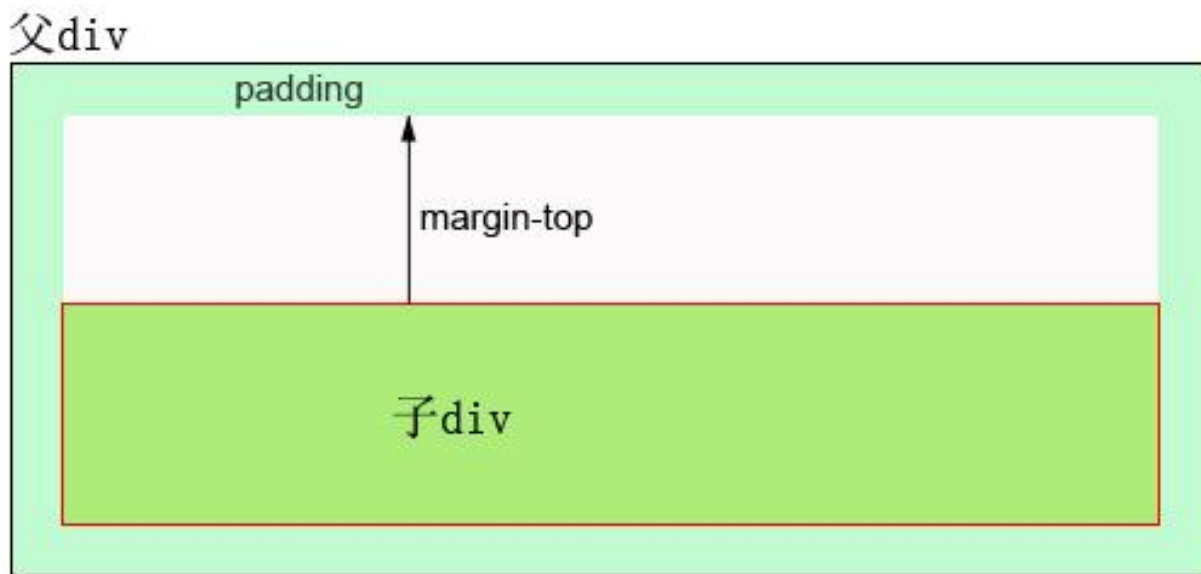


两个块级元素之间的距离**不是**margin-bottom与margin-top总和，**而是**两者中的较大者。

第三章 盒子模型_盒子间的关系



- 嵌套盒子之间的距离 (高度自适应)



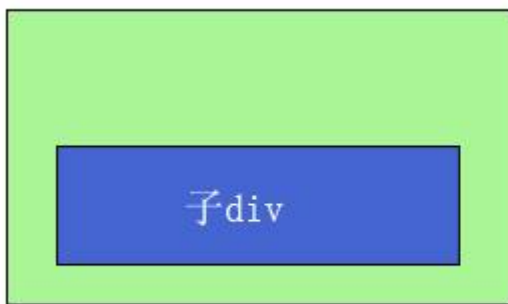
两个嵌套盒子上边框的距离为: $\text{padding} + \text{margin-top}$

第三章 盒子模型__盒子间的关系

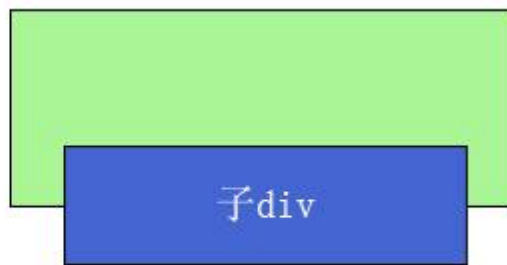


- 嵌套盒子之间的距离(高度固定)

若父div高度，小于子div的“高度+margin-top”值，此时IE浏览器会自动扩大，保持子元素的margin-bottom的空间以及元素自身的padding-bottom；而Firefox就不会，它会保证父元素的高度高度的完全吻合，而这时子元素将超过父元素的范围。



IE浏览器



火狐浏览器

从CSS的标准规范来说，IE的这种处理方法是不合规范的。

第三章 盒子模型__盒子间的关系



- Margin可设置为负值

当margin设为负数时，会使被设为负数的块向相反的方向移动，甚至覆盖在另外的块上。



第四章 CSS文字样式



- 使用过任何文字处理软件的用户对文字排版都不会陌生。例如在Word软件 中可以对文字的字体、大小和颜色等各种属性进行设置。CSS同样可以对HTML页面中的文字进行全面设置。
- 请认真学习以下属性！

第四章 CSS文字样式



属性	描述
font	简写属性。作用是把所有针对字体的属性设置在一个声明中。
font-family	设置字体系列。
font-size	设置字体的尺寸。
font-size-adjust	当首选字体不可用时，对替换字体进行智能缩放。 (CSS2.1 已删除该属性。)
font-stretch	对字体进行水平拉伸。(CSS2.1 已删除该属性。)
font-style	设置字体风格。
font-variant	以小型小写字体或者正常字体显示文本。
font-weight	设置字体的粗细。

第四章 CSS文字样式



font-size 属性可设置字体的尺寸。

值	描述
xx-small x-small small medium large x-large xx-large	把字体的尺寸设置为不同的尺寸，从 xx-small 到 xx-large。 默认值：medium。
smaller	把 font-size 设置为比父元素更小的尺寸。
larger	把 font-size 设置为比父元素更大的尺寸。
length	把 font-size 设置为一个固定的值。
%	把 font-size 设置为基于父元素的一个百分比。

第四章 CSS文字样式



font-style 属性定义字体的风格。

值	描述
normal	默认。浏览器显示一个标准的字体。
italic	浏览器会显示一个斜体的字体。
oblique	浏览器会显示一个倾斜的字体。

第四章 CSS文字样式



font-variant 属性设置小型大写字母的字体显示文本，这意味着所有的小写字母均会被转换为大写，但是所有使用小型大写字母的字母与其余文本相比，其字体尺寸更小。

值	描述
normal	默认。浏览器会显示一个标准的字体。
small-caps	浏览器会显示小型大写字母的字体。

第四章 CSS文字样式



font-weight 属性设置文本的粗细。

值	描述
normal	默认。定义标准的字符。
bold	定义粗体字符。
bolder	定义更粗的字符。
lighter	定义更细的字符。
100 200 300 400 500 600 700 800 900	定义由粗到细的字符。400 等同于 normal，而 700 等同于 bold。

第四章

CSS文字样式——font



值	描述
font-style font-variant font-weight font-size/line-height font-family	设置针对字体的属性。line-height 值可设置行间的空白，此值可以是一个数字、一个百分比或者一个字号。 默认值：取决于浏览器。
caption	定义被有标题的控件使用的字体（比如按钮、下拉列表等）。
icon	定义被图标标记使用的字体。
menu	定义被下拉列表使用的字体。
message-box	定义被对话框使用的字体。
small-caption	
status-bar	定义被窗口状态栏使用的字体。

第五章

CSS文本样式



属性	描述
color	设置文本颜色
direction	设置文本方向。
line-height	设置行高。
letter-spacing	设置字符间距。
text-align	对齐元素中的文本。
text-decoration	向文本添加修饰。
text-indent	缩进元素中文本的首行。
text-shadow	设置文本阴影。CSS2 包含该属性，但是 CSS2.1 没有保留该属性。
text-transform	控制元素中的字母。
unicode-bidi	设置文本方向。
white-space	设置元素中空白的处理方式。
word-spacing	设置字间距。

第五章 CSS文本样式



Direction: 设置文本方向。

值	描述
ltr	默认。文本方向从左到右。
rtl	文本方向从右到左。

第五章 CSS文本样式



`line-height` 属性设置行间的距离（行高）。

值	描述
normal	默认。设置合理的行间距。
number	设置数字，此数字会与当前的字体尺寸相乘来设置行间距。
length	设置固定的行间距。
%	基于当前字体尺寸的百分比行间距。

第五章

CSS文本样式



letter-spacing 属性增加或减少字符间的空白（字符间距）。

值	描述
normal	默认。规定字符间没有额外的空间。
length	定义字符间的固定空间（允许使用负值）。

word-spacing 属性增加或减少单词间的空白（即字间隔）。

值	描述
normal	默认。定义单词间的标准空间。
<i>length</i>	定义单词间的固定空间。

第五章

CSS文本样式



`text-align` 属性规定元素中的文本的水平对齐方式。

值	描述
left	把文本排列到左边。默认值：由浏览器决定。
right	把文本排列到右边。
center	把文本排列到中间。
justify	实现两端对齐文本效果。

民间偏方(两行得同时写上):

```
text-align:justify;
```

```
text-justify:inter-ideograph;
```

第五章 CSS文本样式



`text-decoration` 属性规定添加到文本的修饰。

值	描述
<code>none</code>	默认。定义标准的文本。
<code>underline</code>	定义文本下的一条线。
<code>overline</code>	定义文本上的一条线。
<code>line-through</code>	定义穿过文本下的一条线。
<code>blink</code>	定义闪烁的文本。（不支持IE6）

第五章 CSS文本样式



`text-indent` 属性规定文本块中首行文本的缩进。

值	描述
<i>length</i>	定义固定的缩进。默认值：0。
<i>%</i>	定义基于父元素宽度的百分比的缩进。

常用：

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

第五章 CSS文本样式



`text-transform` 属性控制文本的大小写。

值	描述
none	默认。定义带有小写字母和大写字母的标准文本。
capitalize	文本中的每个单词以大写字母开头。
uppercase	定义仅有大写字母。
lowercase	定义无大写字母，仅有小写字母。

第五章

CSS文本样式



white-space 属性设置如何处理元素内的空白。

值	描述
normal	默认。空白会被浏览器忽略。
pre	空白会被浏览器保留。其行为方式类似 HTML 中的 <code><pre></code> 标签。
nowrap	文本不会换行，文本会在在同一行上继续，直到遇到 <code>
</code> 标签为止。
pre-wrap	保留空白符序列，但是正常地进行换行。
pre-line	合并空白符序列，但是保留换行符。

第六章 背景属性



属性	描述
background	简写属性，作用是将背景属性设置在一个声明中。
background-attachment	背景图像是否固定或者随着页面的其余部分滚动。
background-color	设置元素的背景颜色。
background-image	把图像设置为背景。
background-position	设置背景图像的起始位置。
background-repeat	设置背景图像是否及如何重复。

第六章 背景属性



`background-image` 属性为元素设置背景图像。

值	描述
<code>url (' URL ')</code>	指向图像的路径。
<code>none</code>	默认值。不显示背景图像。

第六章 背景属性



`background-repeat` 属性设置是否及如何重复背景图像。

默认地，背景图像在水平和垂直方向上重复。

值	描述
<code>repeat</code>	默认。背景图像将在垂直方向和水平方向重复。
<code>repeat-x</code>	背景图像将在水平方向重复。
<code>repeat-y</code>	背景图像将在垂直方向重复。
<code>no-repeat</code>	背景图像将仅显示一次。

第六章 背景属性



`background-position` 属性设置背景图像的起始位置。

值	描述
<code>top left</code> <code>top center</code> <code>top right</code> <code>center left</code> <code>center center</code> <code>center right</code> <code>bottom left</code> <code>bottom center</code> <code>bottom right</code>	如果您仅规定了一个关键词，那么第二个值将是“center”。 默认值：0% 0%。
<code>x% y%</code>	第一个值是水平位置，第二个值是垂直位置。 左上角是 0% 0%。右下角是 100% 100%。 如果您仅规定了一个值，另一个值将是 50%。
<code>xpx ypx</code>	第一个值是水平位置，第二个值是垂直位置。 左上角是 0 0。单位是像素 (0px 0px) 或任何其他 CSS 单位。 如果您仅规定了一个值，另一个值将是50%。 您可以混合使用 % 和 position 值。

第六章 背景属性



`background-attachment` 属性设置背景图像是否固定或者随着页面的其余部分滚动。

值	描述
<code>scroll</code>	默认值。背景图像会随着页面其余部分的滚动而移动。
<code>fixed</code>	当页面的其余部分滚动时，背景图像不会移动。

第六章 背景属性



Background简写属性:

值	描述
background-color	您可以在此声明中声明1到5个背景属性 默认值：未定义。
background-image	
background-repeat	
background-attachment	
background-position	

```
background:red url(img.jpg) no-repat center fixed ;
```

第七章 盒子的浮动与定位



- CSS 有三种基本的定位机制：普通流、浮动和定位。
- 除非专门指定，否则所有框都在普通流中定位。也就是说，普通流中的元素的位置由元素在HTML中的位置决定。
- 在下面的章节，我们会为您详细讲解：
浮动、定位

第七章 盒子的浮动与定位



• 浮动

- 浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。
- 由于浮动框不在文档的普通流中，所以文档的普通流中的块框表现得就像浮动框不存在一样。

第七章 盒子的浮动与定位



float 属性定义元素在哪个方向浮动。以往这个属性总应用于图像，使文本围绕在图像周围，不过在 **CSS** 中，任何元素都可以浮动。浮动元素会生成一个块级框，而不论它本身是何种元素。

如果浮动某元素，则要指定一个明确的宽度；否则，它们会尽可能地窄。

值	描述
left	图像或文本浮动在父元素的左边。
right	图像或文本浮动在父元素的右边。
none	图像或文本浮动显示在它在父元素中出现的位置。

第七章 盒子的浮动与定位



- 准备代码, 如图所示:

大公子

二公子

三公子

"执子之手，与子偕老" 谁，执我之手，敛我半世癫狂； 谁，吻我之眸，遮我半世流离；
谁，抚我之面，慰我半世哀伤； 谁，携我之心，融我半世冰霜； 谁，扶我之肩，驱我一世
沉寂。 谁，唤我之心，掩我一生凌轹。 谁，弃我而去，留我一世独殇； 谁，可明我意，使
我此生无憾； 谁，可助我臂，纵横万载无双； 谁，可倾我心，寸土恰似虚弥； 谁，可葬
吾怆，笑天地虚妄，吾心狂。 安！

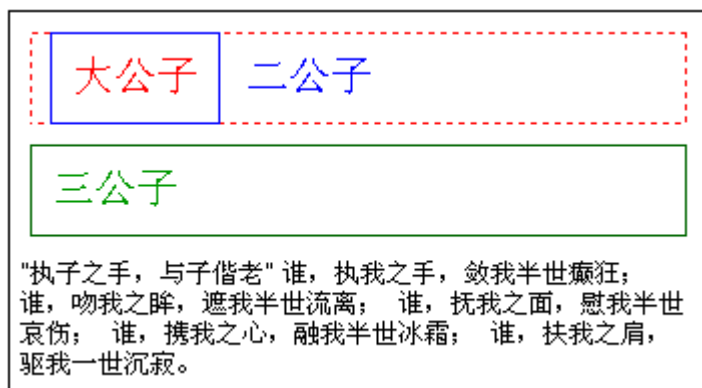
这是标准流布局方式，下面进行浮动试验。

第七章 盒子的浮动与定位

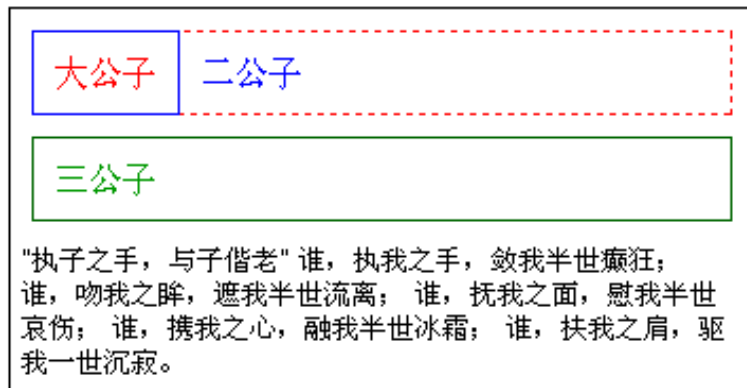


- 试验1：对第一个盒子进行左浮动。

IE浏览器



火狐浏览器



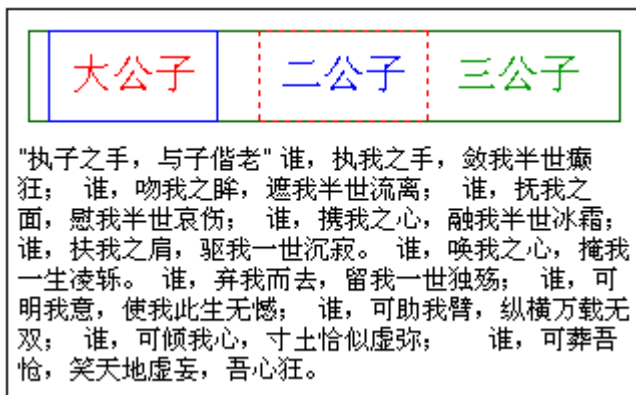
对“大公子”进行浮动后，“大公子”脱离了标准流，“二公子”占据了“大公子”的位置，视“大公子”于无形。

第七章 盒子的浮动与定位

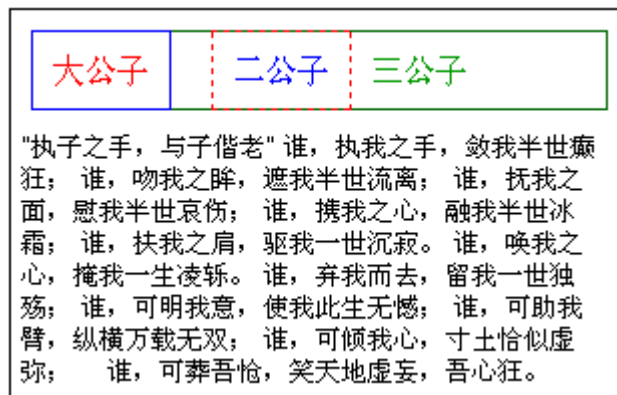


- 试验2：对**第一**和**第二**个盒子进行左浮动。

IE浏览器



火狐浏览器



对“大公子”和“二公子”都进行浮动，它们都脱离了标准流，“三公子”视大哥二哥于无形，占据了它们的位置。

第七章 盒子的浮动与定位



- 试验3：对三个盒子都进行左浮动。

大公子	二公子	三公子	"执子之手，与子偕老" 谁，执我之手，敛我半世癫狂； 谁，吻我之眸，遮我半世流离； 谁，抚我之面，慰我半世哀伤； 谁，携我之心，融我半世冰霜； 谁，扶我之肩，驱我一世沉寂。 谁，唤我之心，掩我一生凌轹。 谁，弃我而去，留我一世独殇； 谁，可明我意，使我此生无憾； 谁，可助我臂，纵横万载无双； 谁，可倾我心，寸土恰似虚弥； 谁，可葬吾怆，笑天地虚妄，吾心狂。 安！
-----	-----	-----	---

对三个“公子”都进行浮动，它们都脱离了标准流，文字部分马上占据了“三个公子”的位置。三个盒子有一定的空间把文字挤到一边，形成图文混排效果。

第七章 盒子的浮动与定位



- 试验4：在例3基础上，对“三公子”右浮动

大公子	二公子	"执子之手，与子偕老" 谁，执我之手，敛我半世癫狂； 谁，吻我之眸，遮我半世流离； 谁，抚我之面，慰我半世哀伤； 谁，携我之心，融我半世冰霜； 谁，扶我之肩，驱我一世沉寂。 谁，唤我之心，掩我一生凌轹。 谁，弃我而去，留我一世独殇； 谁，可明我意，使我此生无憾； 谁，可助我臂，纵横万载无双； 谁，可倾我心，寸土恰似虚弥； 谁，可葬吾怆，笑天地虚妄，吾心狂。 安！	三公子
-----	-----	--	-----

对“三公子”进行右浮动后，盒子进行相应移动，文字依然见缝插针，对盒子进行环绕。

第七章 盒子的浮动与定位



- 试验5: “二公子”和“三公子”浮动互换

大公子	三公子	"执子之手，与子偕老" 谁，执我之手，敛我半世癫狂； 谁，吻我之眸，遮我半世流离； 谁，抚我之面，慰我半世哀伤； 谁，携我之心，融我半世冰霜； 谁，扶我之肩，驱我一世沉寂。 谁，唤我之心，掩我一生凌轹。 谁，弃我而去，留我一世独殇； 谁，可明我意，使我此生无憾； 谁，可助我臂，纵横万载无双； 谁，可倾我心，寸土恰似虚弥； 谁，可葬吾怆，笑天地虚妄，吾心狂。 安！	二公子
-----	-----	--	-----

这就是传说中的“乾坤大挪移”，**CSS布局**，可以实现在**HTML**不做任何改动的情况下，调换盒子的显示位置。

第七章 盒子的浮动与定位



- 试验6：增加div的高度

大公子
好儿子

二公子

三公子

"执子之手，与子偕老" 谁，执我之手，敛我半世癫狂； 谁，吻我之眸，遮我半世流离； 谁，抚我之面，慰我半世哀伤； 谁，携我之心，融我半世冰霜； 谁，扶我之肩，驱我一世沉寂。 谁，唤我之心，掩我一生凌轹。 谁，弃我而去，留我一世独殇； 谁，可明我意，使我此生无憾； 谁，可助我臂，纵横万载无双； 谁，可倾我心，寸土恰似虚弥； 谁，可葬吾怆，笑天地虚妄，吾心狂。安！

- 增加某一个div高度之后，文字依然环绕盒子。

第七章 盒子的浮动与定位



• 试验7：清除浮动

如果不希望文字环绕盒子，就应对文字清除浮动的影响。

大公子
好儿子

二公子

好儿子
好儿子
好儿子
好儿子
好儿子 好儿子

"执子之手，与子偕老" 谁，执我之手，敛我半世癫狂； 谁，吻我之眸，遮我半世流离； 谁，抚我之面，慰我半世哀伤； 谁，携我之心，融我半世冰霜； 谁，扶我之肩，驱我一世沉寂。 谁，唤我之心，掩我一生凌轹。 谁，弃我而去，留我一世独殇； 谁，可明我意，使我此生无憾； 谁，可助我臂，纵横万载无双； 谁，可倾我心，寸土恰似虚弥； 谁，可葬吾怆，笑天地虚妄，吾心狂。 安！

注意：对clear属性的设置要放到文字所在的盒子里，而不是放到“浮动”的盒子里。

```
.clear{ clear:both;}
```

第七章 盒子的浮动与定位



- 试验8：去掉文字后，要解决高度自适应问题。

大公子

二公子

三公子
三公子
三公子
三公子
三公子
三公子
三公子
三公子
三公子

- 去掉文字后，因为三个盒子都脱离了标准流，所以父DIV视三个“公子”于无形。把高度收回。如图所示。
- **解决方法是：**在三个“公子”脚下加`<div class="clear"></div>`，并对`.clear {clear:both;}`进行清除浮动。

第七章 盒子的浮动与定位



• 定位

static（静态定位）

元素框正常生成。

relative（相对定位）

元素框偏移某个距离。元素仍保持其未定位前的形状，它原本所占的空间仍保留。

absolute（绝对定位）

元素框从文档流完全删除，并相对于其包含块定位。包含块可能是文档中的另一个元素或者是初始包含块。元素原先在正常文档流中所占的空间会关闭，就好像元素原来不存在一样。

fixed（固定定位）

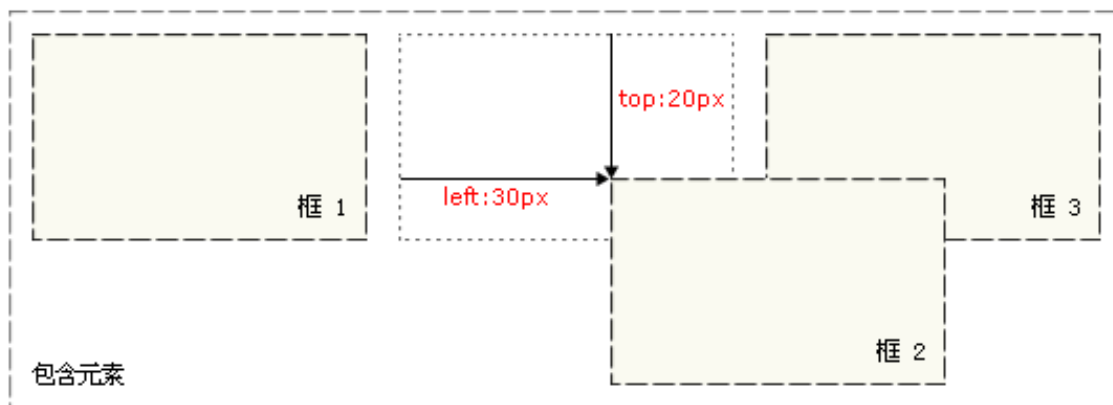
元素框的表现类似于将 **position** 设置为 **absolute**，其包含块是视窗本身。
(不支持IE6)

第七章 盒子的浮动与定位



- CSS 相对定位
- 相对定位是一个非常容易掌握的概念。如果对一个元素进行相对定位，它将以起点为参照，垂直或水平移动。

如下图所示：



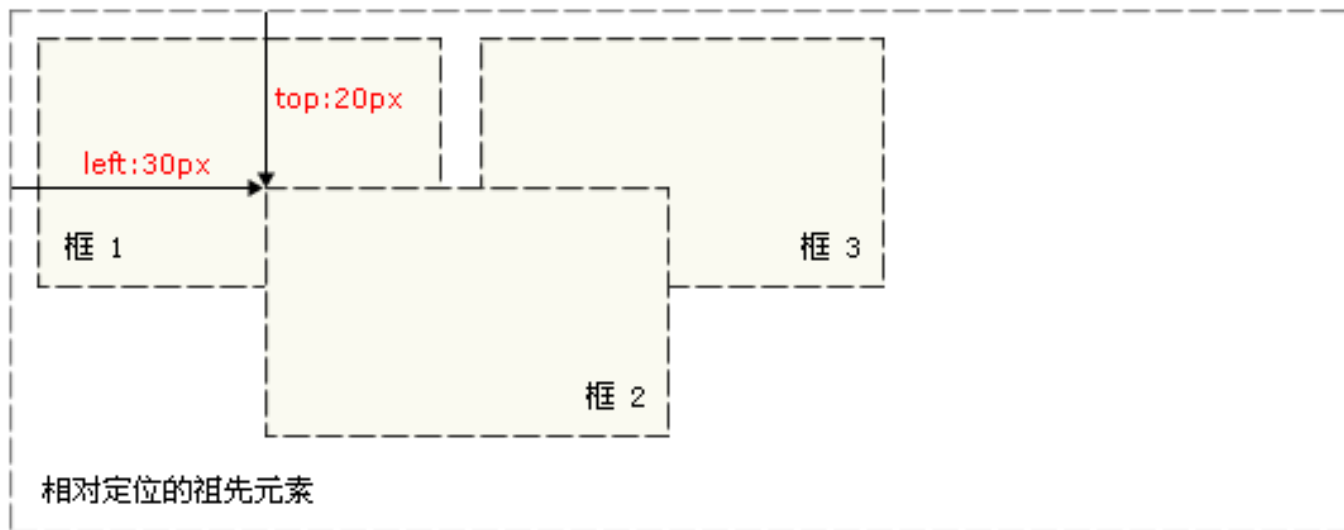
注意：在使用相对定位时，无论是否进行移动，元素仍然占据原来的空间。因此，移动元素会导致它覆盖其它框。

第七章 盒子的浮动与定位



- CSS 绝对定位
- 绝对定位使元素的位置与文档流无关，因此不占据空间。
- 普通流中其它元素的布局就像绝对定位的元素不存在一样。

如下图所示：



第七章 盒子的浮动与定位



属性	描述
position	把元素放置到一个静态的、相对的、绝对的、或固定的位置中。
top	定义了一个定位元素的上外边距边界与其包含块上边界之间的偏移。
right	定义了定位元素右外边距边界与其包含块右边界之间的偏移。
bottom	定义了定位元素下外边距边界与其包含块下边界之间的偏移。
left	定义了定位元素左外边距边界与其包含块左边界之间的偏移。
overflow	设置当元素的内容溢出其区域时发生的事情。
clip	设置元素的形状。元素被剪入这个形状之中，然后显示出来。
vertical-align	设置元素的垂直对齐方式。
z-index	设置元素的堆叠顺序。

第七章 盒子的浮动与定位



`position` 属性规定元素的定位类型。

值	描述
<code>absolute</code>	生成绝对定位的元素，相对于 <code>static</code> 定位以外的第一个父元素进行定位。元素的位置通过 <code>"left"</code> , <code>"top"</code> , <code>"right"</code> 以及 <code>"bottom"</code> 属性进行规定。
<code>fixed</code>	生成绝对定位的元素，相对于浏览器窗口进行定位。元素的位置通过 <code>"left"</code> , <code>"top"</code> , <code>"right"</code> 以及 <code>"bottom"</code> 属性进行规定。
<code>relative</code>	生成相对定位的元素，相对于其正常位置进行定位。因此， <code>"left:20"</code> 会向元素的 <code>LEFT</code> 位置添加 20 像素。
<code>static</code>	默认值。没有定位，元素出现在正常的流中（忽略 <code>top</code> , <code>bottom</code> , <code>left</code> , <code>right</code> 或者 <code>z-index</code> 声明）。

第七章 盒子的浮动与定位



overflow 属性规定当内容溢出元素框时发生的事情。

值	描述
visible	默认值。内容不会被修剪，会呈现在元素框之外。
hidden	内容会被修剪，并且其余内容是不可见的。
scroll	内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。
auto	如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。

第七章 盒子的浮动与定位



vertical-align 属性设置元素的垂直对齐方式。

值	描述
baseline	默认。元素放置在父元素的基线上。
sub	垂直对齐文本的下标。
super	垂直对齐文本的上标
top	把元素的顶端与行中最高元素的顶端对齐
text-top	把元素的顶端与父元素字体的顶端对齐
middle	把此元素放置在父元素的中部。
bottom	把元素的顶端与行中最低的元素顶端对齐。
text-bottom	把元素的底端与父元素字体的底端对齐。
length	
%	使用“line-height”属性的百分比值来排列此元素。允许使用负值。

第七章 盒子的浮动与定位



z-index 属性设置元素的堆叠顺序。拥有更高堆叠顺序的元素总是会处于堆叠顺序较低的元素的前面。

值	描述
auto	默认。堆叠顺序与父元素相等。
<i>number</i>	设置元素的堆叠顺序。

第八章 列表属性与链接



- 从某种意义上讲，不是描述性的文本的任何内容都可以认为是列表。人口普查、太阳系、家谱、参观菜单，甚至你的所有朋友都可以表示为一个列表或者是列表的列表。
- 由于列表如此多样，这使得列表相当重要，所以说，CSS 中列表样式不太丰富确实是一大憾事。

第八章 列表属性与链接



属性	描述
<code>list-style</code>	简写属性。用于把所有用于列表的属性设置于一个声明中。
<code>list-style-image</code>	将图象设置为列表项标志。（常用背景图替代它）
<code>list-style-position</code>	设置列表中列表项标志的位置。
<code>list-style-type</code>	设置列表项标志的类型。
<code>marker-offset</code>	

第八章 列表属性与链接



`list-style-position` 属性设置在何处放置列表项标记。

值	描述
inside	列表项目标记放置在文本以内，且环绕文本根据标记对齐。
outside	默认值。保持标记位于文本的左侧。列表项目标记放置在文本以外，且环绕文本不根据标记对齐。

第八章 列表属性与链接



`list-style-type` 属性设置列表项标记的类型。

值	描述
<code>none</code>	无标记。
<code>disc</code>	默认。标记是实心圆。
<code>circle</code>	标记是空心圆。
<code>square</code>	标记是实心方块。
<code>decimal</code>	标记是数字。
<code>decimal-leading-zero</code>	0开头的数字标记。(01, 02, 03, 等。)
<code>lower-roman</code>	小写罗马数字(i, ii, iii, iv, v, 等。)
<code>upper-roman</code>	大写罗马数字(I, II, III, IV, V, 等。)
<code>lower-alpha</code>	小写英文字母The marker is lower-alpha (a, b, c, d, e, 等。)
<code>upper-alpha</code>	大写英文字母The marker is upper-alpha (A, B, C, D, E, 等。)
<code>lower-greek</code>	小写希腊字母(alpha, beta, gamma, 等。)
<code>lower-latin</code>	小写拉丁字母(a, b, c, d, e, 等。)
<code>upper-latin</code>	大写拉丁字母(A, B, C, D, E, 等。)

第八章 列表属性与链接



list-style 属性是用于在一个声明中设置一个列表的所有属性的简写属性。

值	描述
<code>list-style-type</code> <code>list-style-position</code> <code>list-style-image</code>	设置列表的属性。默认值：未定义。

第八章 列表属性与链接



- 超链接是网页上最普通不过的元素，通过超链接能实现页面的跳转、功能的激活等。
- 在html语言中，超链接通过<a>来实现，链接地址是利用href属性。
- ``
- 上面这种最基本的超链接样式已经无法满足广大设计师的需求。通过CSS可以设置超链接的各种属性。
- `a{ text-decoration:none; color:red;}`

第八章 列表属性与链接



属性	说明
a:link	超链接的普通样式，即正常浏览状态的样式
a:visited	被点击过的超链接的样式
a:hover	鼠标指针经过超链接上时的样式
a:active	在超链接上单击时，即“当前激活”时，超链接的样式。

1、声明的顺序

```
a:link {color: #FF0000} /* unvisited link */  
a:visited {color: #00FF00} /* visited link */  
a:hover {color: #FF00FF} /* mouse over link */  
a:active {color: #0000FF} /* selected link */
```

提示：在 CSS 定义中，a:hover 必须被置于 a:link 和 a:visited 之后，才是有效的。

提示：在 CSS 定义中，a:active 必须被置于 a:hover 之后，才是有效的。

2、在实际工作中，经常使用：a, a:visited{ text-decoration:none;}

3、通常无需设置 a:active 的样式，因为它一闪即逝。

第九章 导航菜单



- 导航是网页中不可缺少的功能，导航菜单充分利用了CSS的各种特性。
- 需要注意的是，导航中使用了前面章节中的CSS盒子模型以及浮动和定位的技术，因此同学们务必先把前面的基础掌握好，再来学习这些实例的案例。

第九章 导航菜单



- 导航（1）：竖排菜单

详见“导航实例1.jpg”



第九章 导航菜单



- 横排菜单

详见“导航实例2.jpg”

[优惠套餐](#)

[域名注册](#)

[虚拟主机](#)

[数据库](#)

[服务器租用](#)

[服务器托管](#)

[网站推广](#)

[企业邮箱](#)

[购物车](#)

[网站地图](#)

第十章 布局元素处理



表格常用来处理新闻：

new 新闻中心		>>更多
» 轴承的极限转速和摩擦系数轴承的注意	2008-08-14	
» 直线轴承LM 与 LME 代号基本含义轴承的注意	2008-08-14	
» SKF润滑脂选择表轴承的注意	2008-08-14	
» 伪劣轴承的危害轴承的注意	2008-08-14	
» 运转中轴承的注意事项：倾听轴承的注意	2008-08-14	

第十章 布局元素处理



UI列表常用来处理这样的循环。

重点推荐

更多>>



经典三粒扣纯色西装 黑色
市场价¥599元 售价¥299元



优雅双下摆小西装
市场价¥599元 售价¥299元



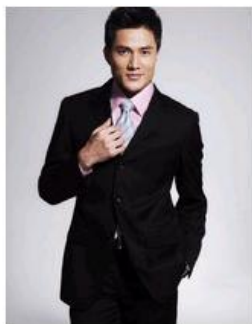
经典三粒扣纯色西装 深蓝色
市场价¥599元 售价¥299元



时尚细条纹西装 黑色
售价¥399元 特惠价¥299元



时尚细条纹西装 蓝黑色
售价¥399元 特惠价¥299元



雅致羊毛条纹西装 翰墨黑
市场价¥1299元 售价¥649元



雅致羊毛条纹西装 深海蓝
市场价¥1299元 售价¥649元



精致暗纹西装 黑色
市场价¥799元 售价¥399元

第十章 布局元素处理

- 元素的命名规范，见（[css样式命名总结.xls](#)）
- 大型门户站css样式命名和应用原则（参考）



作业：

- 1、做完w3school-自学最佳教程上的CSS的70个例子。
- 2、有问题可以在群里问，也可以在网上查询。
- 4、独立完成两个页面的布局
 - 1) 第八章文件夹中的“首页”
 - 2) 第十章中的页面二选一



本课结束，谢谢！

课件下载地址：www.hifoge.com/css

智搏网联学习群：114808677

网络教程：www.w3school.com.cn

