day02_CSS 学习笔记

CSS 的核心为选择器，我们说要想对某个元素进行一个样式的添加，那么首先得选中该元素。然后通过对文字、图片、背景等属性的知识学习，给元素添加具体的样式，为了让整个页面布局比较合理，还需要掌握盒子模型以及定位等知识。

一、CSS 简单介绍以及基本语法

1.什么是 CSS?

CSS：cascading style sheet 层叠样式表。它是用于控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。这样就使得整个页面看起来更加简洁清爽，同时显示的效果也会更加的绚烂。

批注 [ThinkPad1]：在后面讲解 CSS 选择器的时候我们会用代码案例来诠释”层叠”的具体含义。

2.CSS 的语法规范

CSS 在书写的时候遵循下面的规则：

```
<style type="text/css">
    选择器{
        属性名：属性值;
        属性名：属性值;
        属性名：属性值;
    }
</style>
```

批注 [ThinkPad2]：用来声明 style 标签只能用来存放 css 格式的文件。可以省略不写，默认值就是”text/css”

批注 [ThinkPad3]：后面的内容将会学习到

批注 [ThinkPad4]：它总是以键值对的形式出现。

批注 [ThinkPad5]：此处的分号不能省略，否则所有的样式无效，最后一个属性值后面的分号可以省略。

3.CSS 的注释

我们可以是使用 CSS 的注释来对代码进行描述。

```
/*注释代码*/
```

使用技巧：把光标移动到需要添加的注释的行或者选中多行，使用快捷键：Ctrl+/

批注 [ThinkPad6]：注意：很多同学在学习的时候容易较真儿，CSS 的注释是以/*打头，*/结尾，意味着/*后面还可以有 N 多个*,这些*都将为做注释的内容，直至*后面有个/才会结束。

4.开发人员工具

这里以 Firefox 为例，打开写好的 HTML 文件后，在需要查看的元素上面右击鼠标，选择>>>使用 firebug 查看元素。效果如下图所示：

批注 [ThinkPad7]：使用该功能，我们很容易查看元素的样式以及布局情况，这对于局部内容的样式、布局微调很有帮助。

HTML 页面代码：

```
<body>
    <p class="tieshou">铁手</p>
    <p>冷血</p>
    <p>无情</p>
    <p>追命</p>
</body>
```

但凡页面某个元素没有显示出我们预期的效果，都可以使用 firebug 来查看元素，进行纠错和学习。

5.CSS 的引入方式

CSS 的引入方式有四种，分别为行内样式、内部(内嵌式)、外部(链接式)以及导入式。

批注 [ThinkPad8]: 这些叫法可能不是很标准，但是很容易区分引入的方式，学习起来较为容易的掌握。【很多书籍称为：行内、内嵌、链接、导入】

5.1 行内样式

直接在 HTML 的元素上使用 style 属性编写 CSS。

```
<p style="color: red;font-size: 20px;">冷血</p>
```

注意：这里不能将 CSS 写在 CSS 标签中，必须使用 style 属性来进行指定。

5.2 内部样式(内嵌式)

在<head>标签中使用<style>标签来定义 CSS 属性。

```
<style type="text/css">
    p{
        font-style: italic;/*文本的样式*/
        font-size: 18px;/*文本的大小*/
        font-family:"楷体";/*文本的字体*/
        color: red;/*文本的颜色*/
    }
```

批注 [ThinkPad9]: Type 属性可以不写，默认值为：text/css

批注 [ThinkPad10]: 这是注释内容。

批注 [ThinkPad11]: 文本只有 color 属性，没有 font-color 属性!!

```
</style>
```

5.3 外部样式(链接式)

将 CSS 定义成一个.css 文件，然后引入到 HTML 中。

```
<link type="text/css" rel="stylesheet" href="1.css" />
```

这种方式是使用频率最高的，也是最为实用的方法。它将 HTML 页面与 CSS 文件进行了分离，这样使得前期制作和后期维护非常方便，网站后台技术人员与美工设计者也可以很好的分工合作。

5.4 导入样式

它与链接式的功能基本相同，只是语法和运作方式上略有区别。采用此种方式，在 HTML 文件初始化的时候，会被导入到 HTML 文件内，作为文件的一部分，类似内嵌的效果。而链接式则是在 HTML 的标记需要格式时才以链接的方式引入。

使用下面的方式可以完成导入操作：【放入到<style></style>标记之间】

```
@import url("global.css");
```

5.5 各种方式的优先级问题

如果上面四种方式同时作用到同一个 HTML 文件的同一个标记上时，将会出现优先级问题。如果他们设置了不同的属性，那么会同时生效，如果设置了相同的属性，情况就比较复杂。

```
    <style type="text/css">
        p{
            color: red;/*文本的颜色*/
        }
    </style>
</head>
<body>
    <p style="color: black;font-size: 20px;">冷血</p>
</body>
```

通过上面的代码测试，最终“冷血”显示的颜色为黑色。依次使用这种方式来进行测试，发现了如下规律：

行内样式优先级>内嵌式优先级>链接式优先级>导入式优先级

开发经验：虽然各种 CSS 样式加入到页面的方式有先后的优先级，但在建设网站时，最好只是用其中的 1-2 种，这样既有利于后期的维护和管理，也不会出现各种样式“撞车”的情况，便于设计者理顺设计的整体思路。

6.CSS 选择器

选择器(selector)是 CSS 中很重要的概念，所有 HTML 语言中的标记都是通过不同的

批注 [ThinkPad12]: 用来声明 style 标签只能用来存放 css 格式的文件。

批注 [ThinkPad13]: 指定为样式的脚本

批注 [ThinkPad14]: 指定 CSS 文件的路径(位置)

批注 [ThinkPad15]: 可以将同一个 CSS 文件链接到多个 HTML 文件中去

批注 [ThinkPad16]: 此种方式作为了解即可。

批注 [ThinkPad17]: 为了精确的查找某个元素来设计的。

CSS 选择器进行控制的。用户只需要通过选择器对不同 HTML 标签进行控制，并赋予各种样式声明，即可实现各种效果。

6.1 标签选择器

根据指定的标签名给对应的标签元素设置样式。

```
<style type="text/css">
    标签名{
        属性名1：属性值1;
        属性名2：属性值2;
        属性名3：属性值3;
    }
</style>
```

批注 [ThinkPad18]：用来声明 style 标签只能用来存放 css 格式的文件。可以省略不写，默认值就是”text/css”

批注 [ThinkPad19]：它总是以键值对的形式出现。

批注 [ThinkPad20]：此处的分号不能省略，否则所有的样式无效，最后一个属性值后面的分号可以省略。

缺点：标签是有限的，当我们要对某些相同内容设置不同的样式时，此时将会出现标签不够用的情况。(场景：我们希望对文本内容”ITcast”设置不用的颜色，我们发现使用完、<h1>、 等标签后，没有可用的标签了。。。)

6.2 class(类)选择器

根据类名给对应的标签设置样式

```
<style type="text/css">
    .class名{
        属性名1：属性值1;
        属性名2：属性值2;
        属性名3：属性值3;
    }
</style>
```

批注 [ThinkPad21]：每一个标签都可以定义一个 class 属性，后面的属性值叫做 class 名(类名)

类名，好比人的名字，它可以重复，也就是一个类名可以作用到多个标签上

6.3 id 选择器

根据 id 名给对应的标签设置样式

```
<style type="text/css">
    #id名{
        属性名1：属性值1;
        属性名2：属性值2;
        属性名3：属性值3;
    }
</style>
```

批注 [ThinkPad22]：每一个标签都可以定义一个 id 属性，后面的属性值叫做 id 名

id 名，好比人的身份证号码，它是唯一的(不可以重复)，也就是说一个 id 名只作用到一个标签上。

批注 [ThinkPad23]：虽然在页面中可以用个，但是极力不推荐，会造成文件的紊乱。在后续的课程学习中，但凡涉及到 id 都是为了定一个不重复的标志。

总结：关于类选择器与 id 选择器，我们需要注意一下几点

1.不要在页面上写一些无用的样式(这些样式的选择器不存在或者不正确)

2.一个类名可以作用于多个标签上面，一个标签也有可以有多个类名(多个类名之间用空格隔开)。

3.一个 id 只可以作用于一个标签上面，一个标签也只有一个 id 名。

4.类选择器与 id 选择器在命名规则上面需要注意，只能是英文字母、数字、连接符(-)、下划线(_)，并且数字不能打头。(连接符、下划线打头后面需要跟字母)

7.CSS 的三大特性

批注 [ThinkPad24]: 了解即可，此章节不必深究！

7.1 继承

- 作用：
  给父元素设置一个属性，然后子元素可以使用
- 应用：
  如果页面上有很多的文件都是红色，并且大小都是 20px，那么这个时候给每个元素单独设置会很麻烦，所以可以考虑继承。

```
    <style type="text/css">
        body{
            color: red;
        }
    </style>
</head>
<body>
    <p>
        <span>大娃</span>
    </p>
    <span>
        二娃
    </span>
    <div>
        <span>三娃</span>
    </div>
</body>
```

批注 [ThinkPad25]: 我们希望所有 span 标签的内容显示颜色为红色，单独为每一个设置比较麻烦，此时可以使用其父元素进行设置，所有子元素都会继承！

- 注意：
  将来在写代码的时候，我们的 css 初始化，与页面文字的整体颜色一般会先设置。
- 什么样的属性才可以继承呢？
  凡是以 line-,text-,font-开头的属性都是可以继承。

7.2 层叠

层叠是页面处理冲突属性的一个能力。层叠的最终结果跟优先级有关系。

- 如果多个选择器为同一个元素设置了不同的属性它们会同时作用于这个元素。
- 如果多个选择器为同一个元素调协了相同的属性它们会发生层叠。

7.3 优先级

优先级从大到小

！important>Id>类>标签>通配符>继承>浏览器默认

二、CSS 显示效果之相关属性知识介绍

1.与文本相关的属性

文本相关的属性有：文本的样式、大小、字体、粗细、颜色等

```
p{
    font-style: italic;/*文本的样式*/
    font-size: 18px;/*文本的大小*/
    font-family:"楷体";/*文本的字体*/
    font-weight: 900;/*文本的粗细*/
    color: red;/*文本的颜色*/
}
```

1.1 文本的样式

文本样式使用 font-style 属性来指定，其属性值有 italic(斜体)、normal(正常的)等等

1.2 文本的大小

属性：font-size ， 取值可以为像素值。

1.3 文本的字体

属性：font-family ， 取值为本机所支持的所有字体。汉字的时候需要加引号。

1.4 文本的粗细

属性：font-weight ，取值可以是数字(100-900 的整百数值)，也可以是关键字(bold 加粗，bolder 比加粗更粗，lighter 细线)

1.5 字体属性的连写

格式：

font：font-style font-weight font-size font-family

注意：

这些属性中，font-size 和 font-family 是必写的(必须放置在会后面)，其余的是可选的。

1.6 文本的颜色

属性：color，没有 font-color 属性！

取值：总共有 4 种取值方式

(1)使用具体的颜色值(英文单词)：color：red;

(2)使用十六进制数字取值， color：#FF00FF

(3)使用 rgb 表示法，color：rgb(0,0,0)，三个值分表代表红、绿、蓝，每个取值都是 0-255 或者是百分比。

(4)使用 rgba 表示法，不做介绍！(后端开发人员都这些有个大致了解就可以了)

2.与背景相关的属性

2.1 背景颜色

Background-color：设置背景颜色

2.2 背景图片

Background-image:url(../aaa/bbb/love.jpg);

2.3 是否平铺

Background-repeat:repeat-y

批注 [ThinkPad26]：X:在水平方向进行平铺；Y:在垂直方向进行平铺。

2.4 背景图片的位置

Background-position:center

background-position 属性提供值有很多方法。首先，可以使用一些关键字：top、bottom、left、right 和 center。通常，这些关键字会成对出现，不过也不总是这样。还可以使用长度值，如 100px 或 5cm，最后也可以使用百分数值。不同类型的值对于背景图像的放置稍有差异。

三、CSS 盒子模型

我们在写网站的时候说白了就是通过 div 设置页面，再直白一点就是在页面中码盒子！(查看苹果官网，鼠标定位到具体的一行，就可以看到盒子的雏形)

如果我们要把一个页面写得好看一点，就是要把盒子放得好一点。页面上的每一个元素都是一个盒子！万物皆盒子！

1.盒子的组成

单个盒子

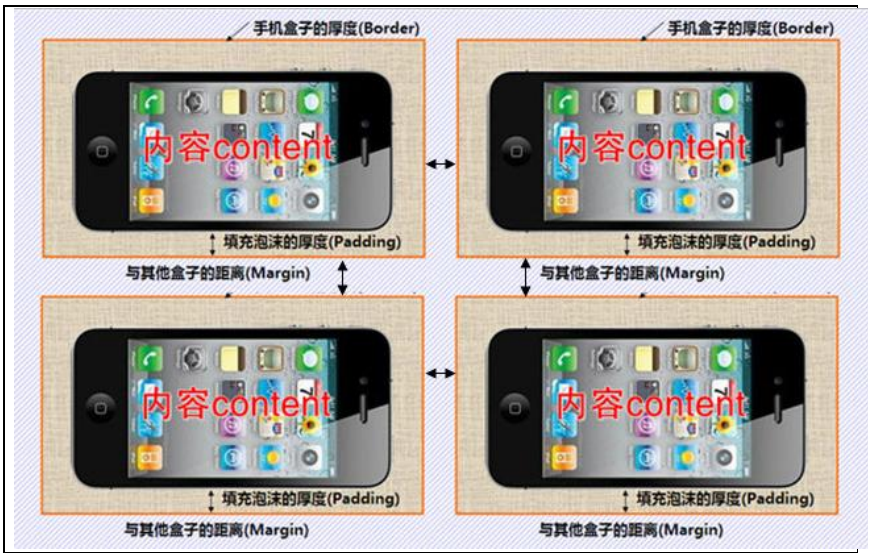
白话解说：

上图中，红色边框为手机的外包装盒，而且外包装盒有一定的厚度(border)，为了保护手机在运送的过程中不被磕坏，我们在盒子里面四周填充了一层泡沫，并且泡沫也有一定的厚度(padding)，这层泡沫将手机包裹住，手机就是整个盒子的核心内容(content)。

可以这么对比记忆：

手机------>>>内容(content)
泡沫------>>>padding
包装盒---->>>boder
外包装间距-->>>margin

多个盒子：



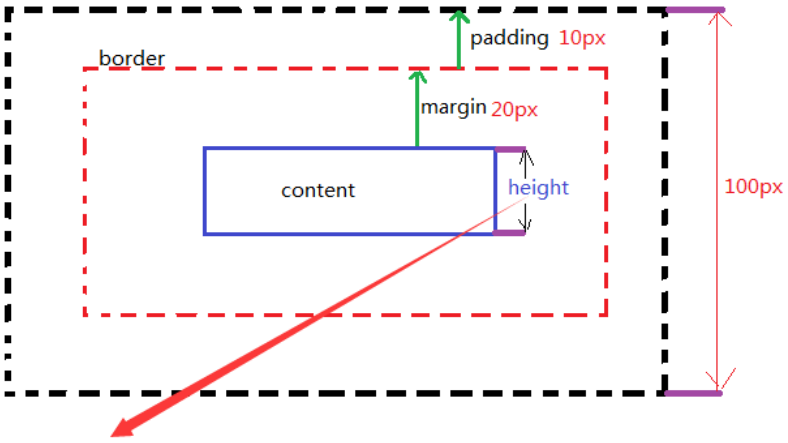
2.盒子取值计算

我们在修改页面元素位置的时候，需要设定它相对于盒子的位置，那么我们有必要清楚关于盒子模型里面的取值相关问题。

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总尺寸。而默认情况下，内边距、边框和外边距的值均为 0.

占据页面大小的区域是整个元素框的总尺寸！默认情况，padding、margin、border 均为 0，假如我们设定了区域内容的 width 和 height，那么此时整个元素框的总尺寸就是该区域内容的宽高了，此时，如果设定了 margin 值，那么整个元素框的总尺寸会发生变化(变大了)，但是我们希望它的整体布局不发生变化！所有我们可以修改区域内容的尺寸(原有大小减去设定的 margin 值)。

附：图解说明盒子模型取值问题



未设置padding和margin的时候，整个元素框的值为内容区域的取值，假定为其height为100px
现在设定其padding为10px，margin为20px
为了保证整个元素框的尺寸不发生变化(不影响整体布局)，那么需要修改内容区域的height值了！
此时内容区域的height值为：原始值100px-padding(10px)*2-margin(20px)*2=**40px**

四、CSS 定位

定位的基本思想很简单，它允许你定义元素框相对于其正常位置应该出现的位置，或者相对于父元素、另一个元素甚至浏览器窗口本身的位置。

CSS 的定位机制：普通流、浮动和绝对定位！如果没有特殊说明，一般都是以普通流的方式进行定位。

块级框从上到下一个接一个的排列。框之间的垂直距离是由框的垂直外边距计算出来。

行内框在一行中水平布置。可以使用水平内边距、边框和外边距调整它们的间距。但是，垂直内边距、边框和外边距不影响行内框的高度。由一行形成的水平框称为*行框（Line Box）*，行框的高度总是足以容纳它包含的所有行内框。不过，设置行高可以增加这个框的高度。

1.相对定位

如果对一个元素进行相对定位，它将出现在它所在的位置上。然后，可以通过设置垂直或水平位置，让这个元素“相对于”它的起点进行移动。

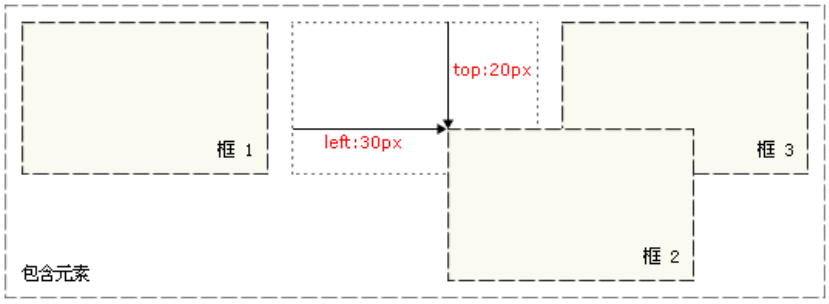
如果将 `top` 设置为 `20px`，那么框将在原位置顶部下面 `20` 像素的地方。如果 `left` 设置为 `30` 像素，那么会在元素左边创建 `30` 像素的空间，也就是将元素向右移动。

代码：

```
#box_relative {
    position: relative;
    left: 30px;
    top: 20px;
}
```

批注 [ThinkPad27]: id 选择器

批注 [ThinkPad28]: 相对定位



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。

2.绝对定位

绝对定位是“相对于”最近的已定位祖先元素，如果**不存在已定位的祖先元素**，那么“相对于”最初的包含块。

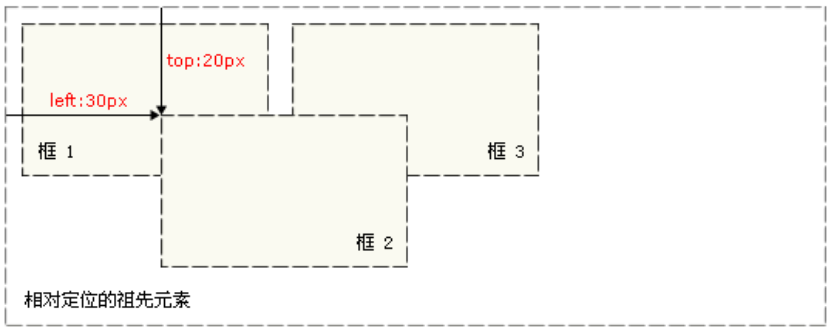
绝对定位使元素的位置与文档流无关，因此不占据空间。这一点与相对定位不同，相对定位实际上被看作普通流定位模型的一部分，因为元素的位置相对于它在普通流中的位置。

代码：

```
#box_relative {
    position: absolute;
    left: 30px;
    top: 20px;
}
```

批注 [ThinkPad29]: id 选择器

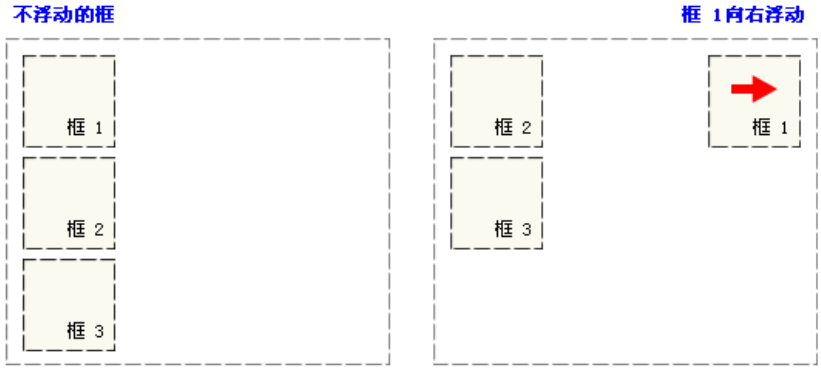
批注 [ThinkPad30]: 绝对定位



3.浮动

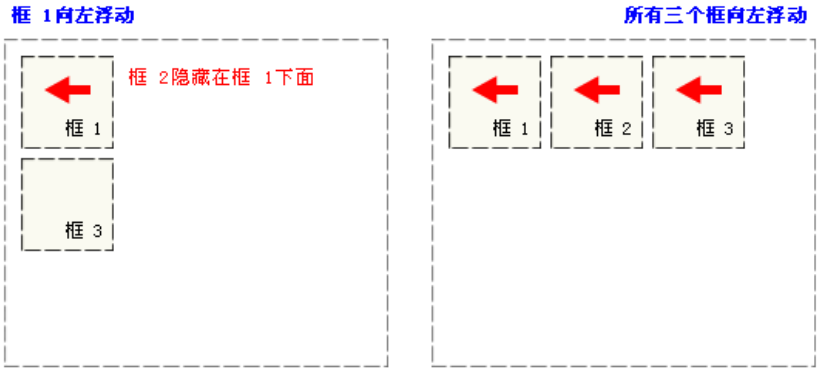
浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。由于浮动框不在文档的普通流中，所以文档的普通流中的块框表现得就像浮动框不存在一样。

请看下图，当把框 1 向右浮动时，它脱离文档流并且向右移动，直到它的右边缘碰到包含框的右边缘：

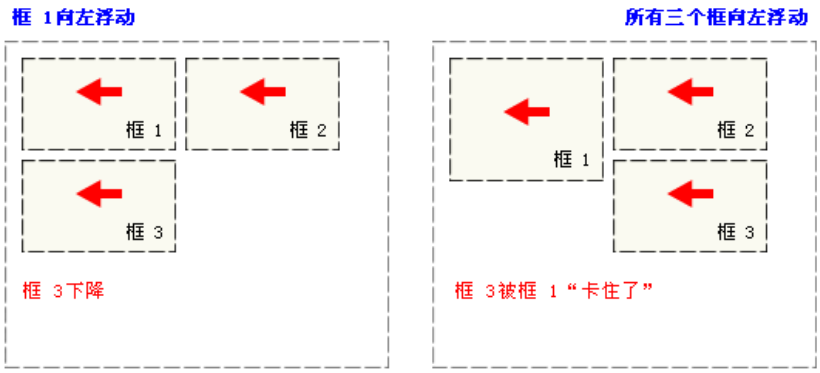


再请看下图，当框 1 向左浮动时，它脱离文档流并且向左移动，直到它的左边缘碰到包含框的左边缘。因为它不再处于文档流中，所以它不占据空间，实际上覆盖住了框 2，使框 2 从视图中消失。

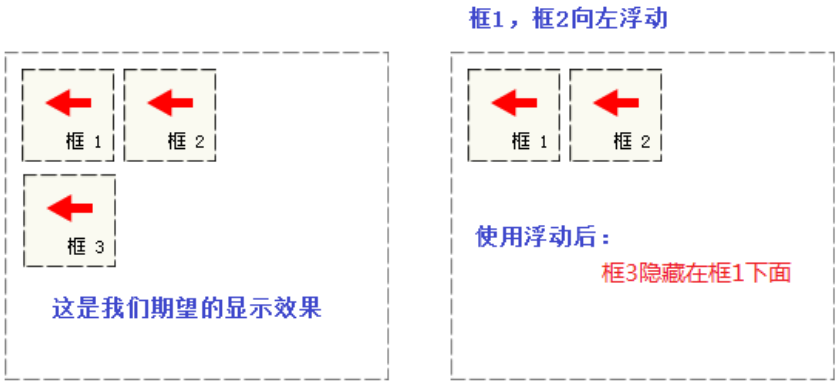
如果把所有三个框都向左移动，那么框 1 向左浮动直到碰到包含框，另外两个框向左浮动直到碰到前一个浮动框。



如下图所示，如果包含框太窄，无法容纳水平排列的三个浮动元素，那么其它浮动块向下移动，直到有足够的空间。如果浮动元素的高度不同，那么当它们向下移动时可能被其它浮动元素“卡住”：



清除浮动



我们要对网站进行布局，显示效果希望是上图左边的效果，此时我们会将框 1 和框 2 向左进行浮动，由于使用了浮动，它们已经**脱离了文档流**，框 3 会上移至原来框 1 的位置，导致的现象是框 3 隐藏在框 1 下面。

那么此时，我们可以清除浮动来清除之前框 1 和框 2 使用浮动后造成的问题！

解决办法：

在框 3 的前面定义一个 div( )

定义 CSS 样式：

```
#three{
    clear:both;
}
```