## 一、布局

### 1. 浮动 float

#### 1.1 浮动的概念

float是css样式中的定位属性，用于设置标签对象（如：<div>标签盒子、<p>标签、<li>标签等html标签）的**浮动布局**，浮动也就是我们所说标签对象浮动居左靠左（float:left）和浮动居右靠右(float:right)。

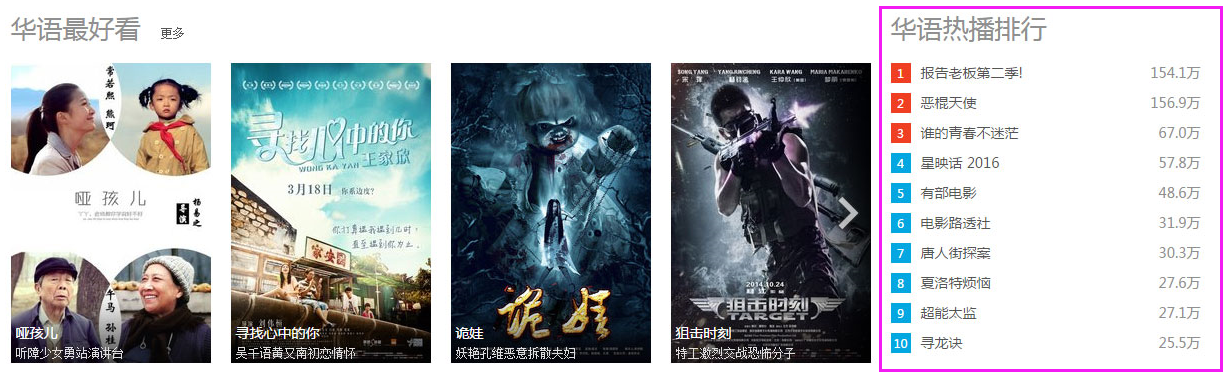
浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。

只要设置了浮动，就将**盒子脱离了文本流**。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。

#### 1.2 浮动的应用



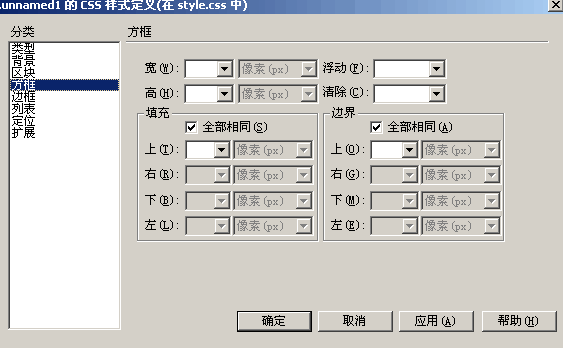
**（左右布局）**



**（左右布局）**



**（左中右布局）**





#### 1.3 浮动属性 float

作用： 该属性的值指出了对象是否及如何浮动。

语法： **float :none**| **left**|**right**

示例：

|  |
| --- |
| /\*左浮动和右浮动\*/ div { **float:left;** }  div { **float:right;** } |

注：

只要设置了浮动，**盒子将脱离普通文本流，下方的内容会自动上移**。解决的办法是：使用clear属性清除浮动带来的效果。

#### 1.4 清除浮动

作用： 清除浮动造成的影响。

语法： **clear :none**| **left**|**right**| **both**

示例：

|  |
| --- |
| div { **clear:left;** } /\* 清除左浮动 \*/  div { **clear:both;** } /\* 清除左右浮动 \*/ |

### 2. 超出 overflow

作用： 检索或设置当对象的内容超过其指定高度及宽度时如何管理内容。

语法： **overflow :visible**| **auto**| **hidden**| **scroll**

示例：

|  |
| --- |
| div { **overflow: hidden;** } /\* 超出部分隐藏 \*/ |

设置textarea对象为hidden值将隐藏其滚动条。

### 3. 显示与隐藏 display

作用： 设置或检索对象是否及如何显示。

语法： **display : block**| **none**|**inline**

示例：

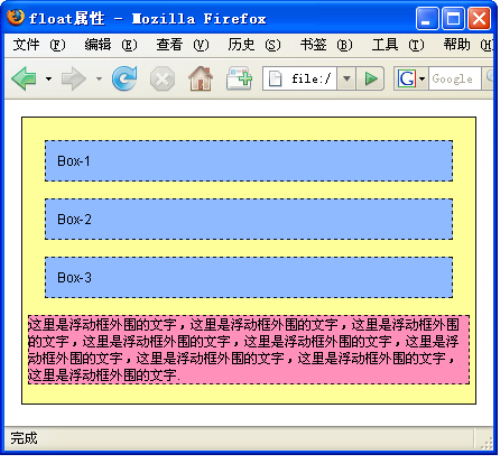
|  |
| --- |
| a { **display: block;** } /\* 以块的方式显示A标签 \*/  div { **display: none;** } /\* 不显示DIV标签 \*/  div { **display: inline;** } /\* 以行内元素的方式显示DIV标签 \*/ |

通常用于制作滑动门等多种网页特效。



### 4 浮动的实例

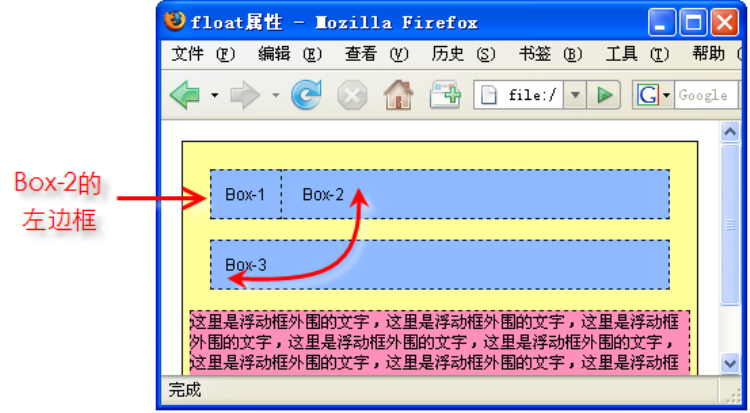
#### 4.1 未浮动的效果



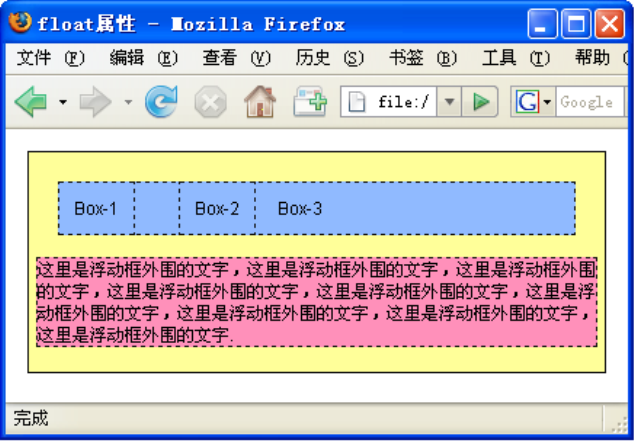
源码如下：

|  |
| --- |
| <!DOCTYPE html>  <html lang="zh-CN">  <head>  <meta charset="utf-8"/>  <title>float属性</title>  <style type="text/css">  html,body{background:#ff9;}  div{margin:10px;border:1px dashed #333;background:#90baff;padding:12px;}  p{border:1px dashed #333;background:#ff90ba;}  </style>  </head>  <body>  <div class="box1">Box-1</div>  <div class="box2">Box-2</div>  <div class="box3">Box-3</div>  <p>这里是浮动的外围文字，这里是浮动的外围文字，这里是浮动的外围文字，这里是浮动的外围文字，这里是浮动的外围文字，这里是浮动的外围文字，这里是浮动的外围文字，这里是浮动的外围文字</p>  </body>  </html> |

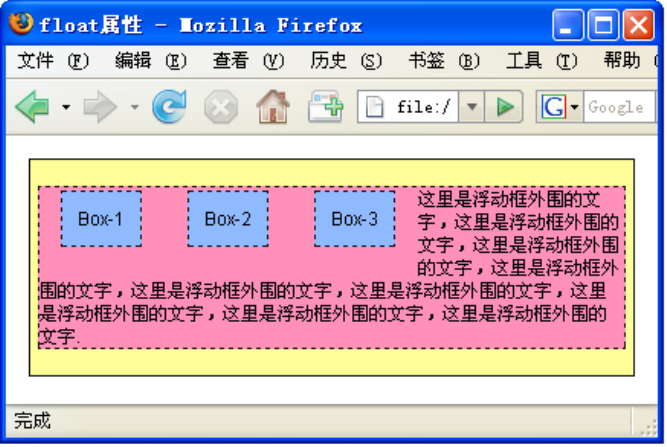
#### 4.2 设置第1个div左浮动时的效果



#### 4.3 设置2个div左浮动时的效果



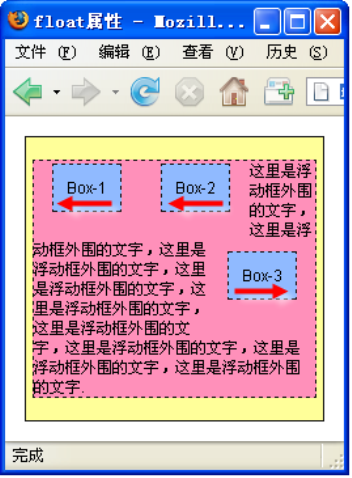
#### 4.4 设置3个div左浮动时的效果



#### 4.5 第3个DIV改变浮动方向后的效果



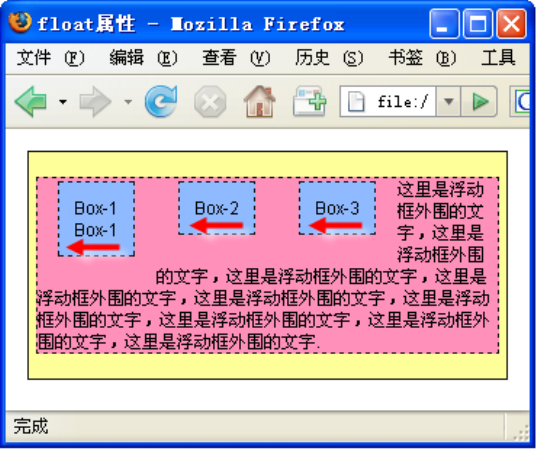
#### 4.6 div被挤到下一行时的效果



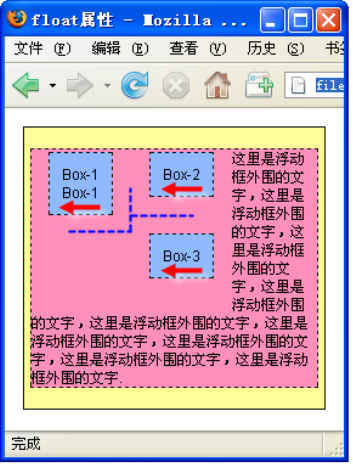
#### 4.7 交换div位置时的效果



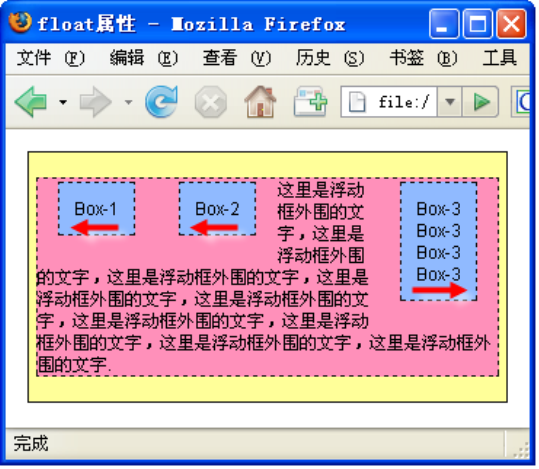
#### 4.8 增加div内容时浮动的效果



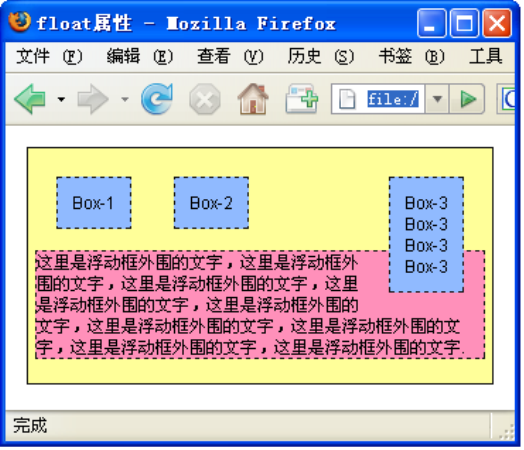
#### 4.9 div挤倒下一行被卡住时的效果



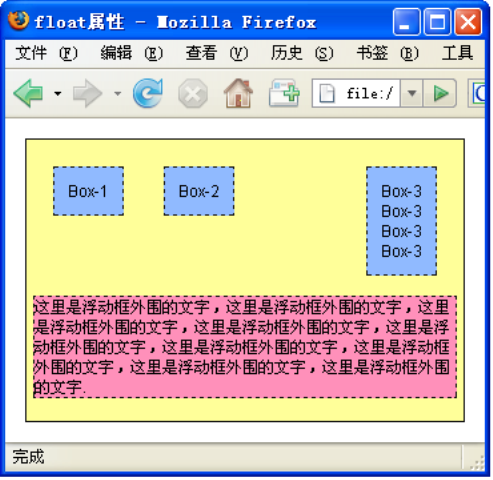
#### 4.10 设置浮动后文字环绕的效果



#### 4.11 清除浮动对左侧影响后的效果



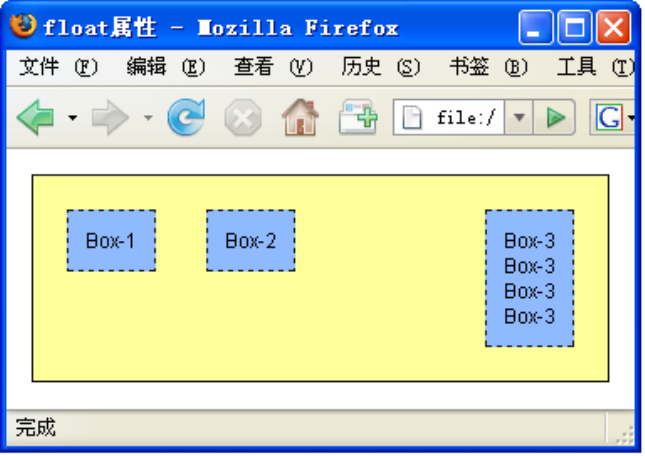
#### 4.12 清除浮动两侧影响后的效果



#### 4.13 包含浮动div的容器将不会适应高度



#### 4.14 希望实现的效果



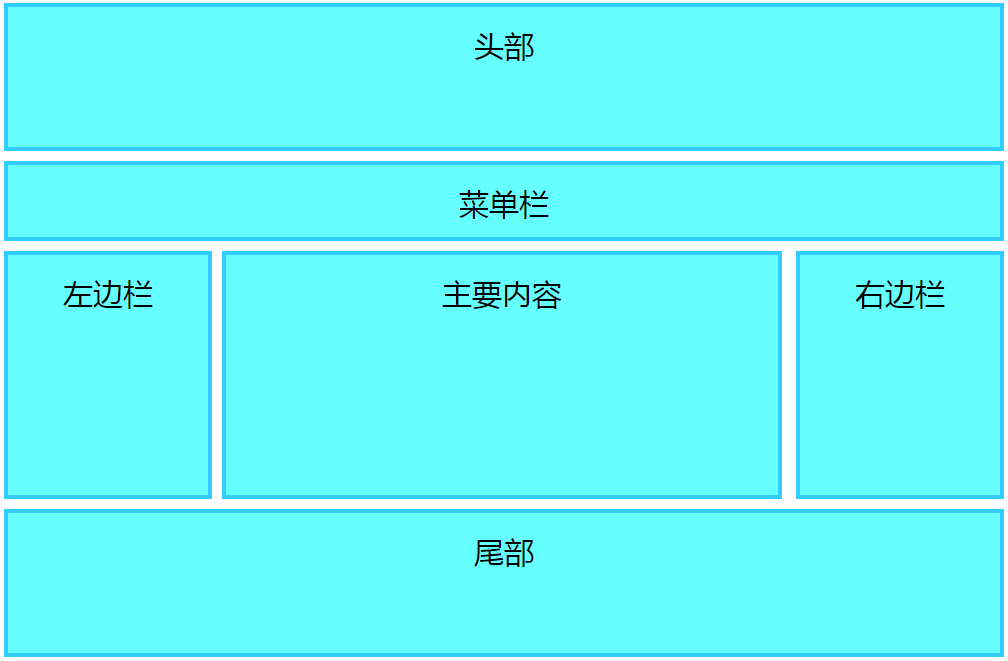
**技巧：**可以使用两种方法实现目标效果：

1. 添加一个空的DIV 设置成清除浮动。
2. 在大的DIV上使用 overflow属性 overflow:hidden 。

清除浮动的3种方式

1. Clear:both
2. overflow:hidden
3. :after伪元素实现元素末尾添加一个看不见的块元素（Block

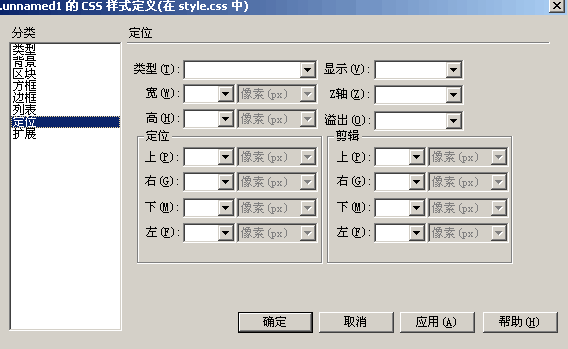
练习：

制作一个网站的基本架构模型，宽度定为 1000px

## 二、定位 position（掌握）

1）广义的“定位”：要将某个元素放到某个位置的时候，这个动作可以称为定位操作，可以使用任何CSS规则来实现，这就是泛指的一个网页排版中的定位操作，使用传统的表格排版时，同样存在定位的问题。

2）狭义的“定位”：在CSS中有一个非常重要的属性position，这个单词翻译为中文也是定位的意思。然而要使用CSS进行定位操作并不仅仅通过这个属性来实现。





### 1. 定位 position

作用： 设置或检索对象的定位方式。

语法： **position :static**| **absolute** | **fixed**| **relative**

示例：

|  |
| --- |
| a { **position: absolute;** } /\* A标签设置成绝对定位 \*/  div { **position: fixed;** } /\* DIV标签设置成固定定位 \*/  div { **position: relative;** } /\* DIV标签设置成相对定位 \*/ |

absolute 和 fixed，将对象从文档流中拖出，配合left/right/top/bottom进行独立定位控制。

### 2. 层次 z-index

作用： 检索或设置对象的层叠顺序。

语法： **z-index :auto**| *number*

示例：

|  |
| --- |
| div {position: absolute; **z-index:999;** } |

**与 position搭配使用**。只有当**DIV定位方式设置成 absolute 或 fixed** 时，该属性才起作用，**层次值高的会遮住层次值低的对象**。越大越在上面

### 3. 位置控制 left right top bottom

作用： 控制标签对象的位置。

语法：

**left :** **auto|***number*

**right:** **auto|***number*

**top:** **auto|***number*

**bottom:** **auto|***number*

示例：

|  |
| --- |
| div {position: absolute; **left:100px;** }  div {position: absolute; **right:100px;** }  div {position: absolute; **top:100px;** }  div {position: absolute; **bottom:100px;** } |

与 position搭配使用。

1）当对象的定位方式是**相对定位**的时，**以当前对象为基准**，表示偏移量。

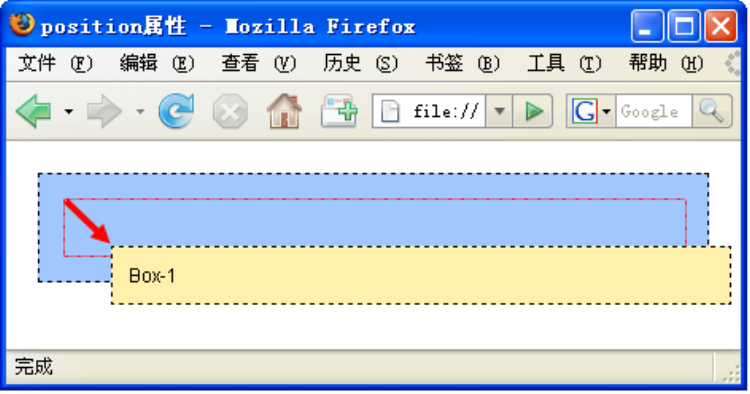
2）当对象的定位方式是**固定定位**时，**以浏览器为基准**，表示与浏览器边框之间的距离。

3）当对象的父对象有设置定位，**以父对象为不是static基准,**，表示与父对象之间的距离。

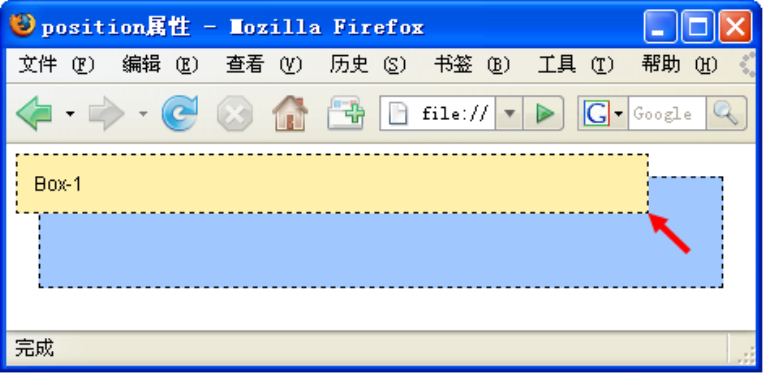
### 4. 定位实例

#### 4.1 相对定位

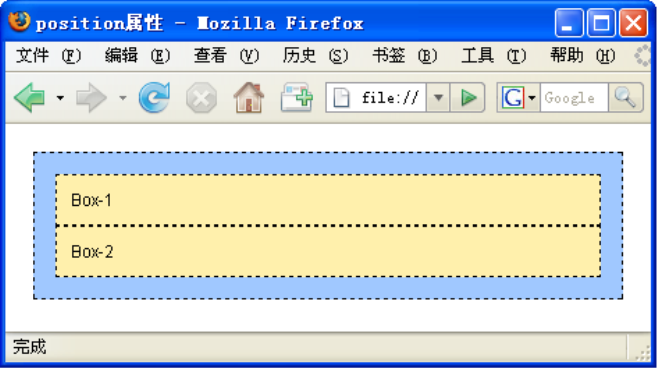
##### 4.1.1一个div设置为相对定位后的效果



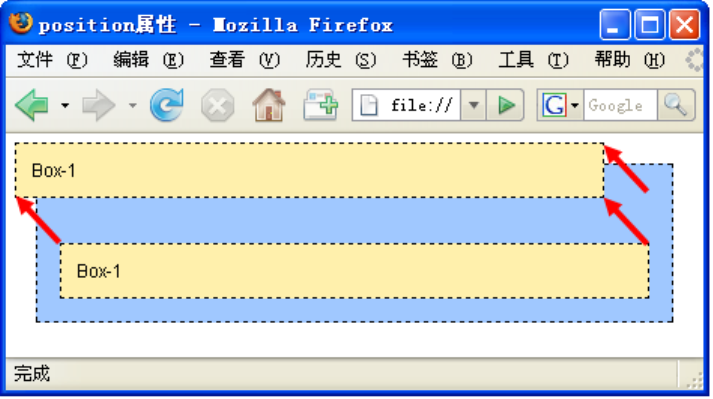
##### 4.1.2 以右侧和下侧为基准设置相对定位



##### 4.1.3 设置两个DIV相对定位的效果



相对定位前



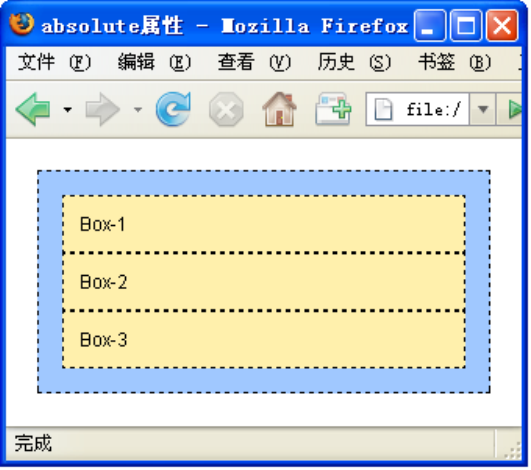
相对定位后

“相对定位”的定位原则：

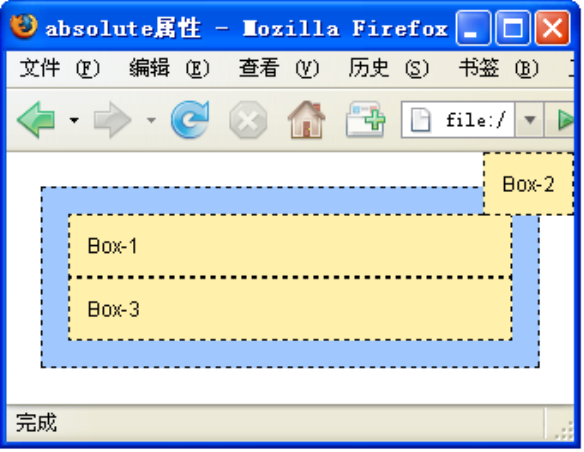
1. **使用相对定位的盒子，会相对于它在原本的位置，通过偏移指定的距离，到达新的位置。**
2. **使用相对定位的盒子仍在标准文本流中，它对父块和兄弟盒子没有任何影响。**

#### 4.2 绝对定位

##### 4.2.1 设置绝对定位前的效果



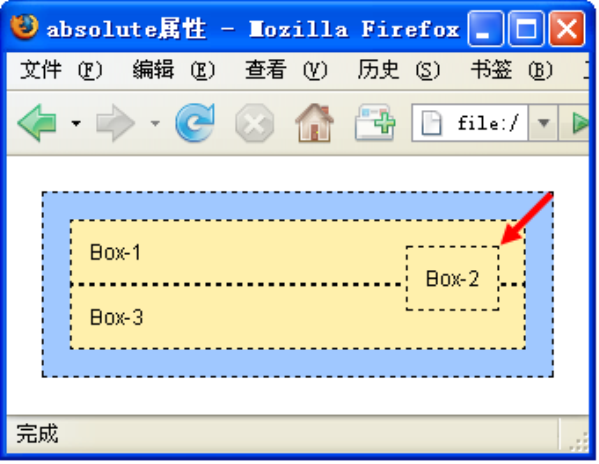
##### 4.2.2 将中间的div设置为绝对定位后的效果



##### 4.2.3 设置偏移量后的效果



##### 4.2.4 将父块设置为“包含块”后的效果



绝对定位的注意事项：

1. 使用绝对定位的盒子以它的**“最近”一个“ 非static的父元素**。如果没有已经定位的父元素，那么会以浏览器窗口为基准进行定位。
2. **绝对定位的框从标准流中脱离**，这意味着它们对其后的兄弟盒子的定位没有影响，其他的盒子就好像这个盒子不存在一样。
3. 关于“**最近**”，在一个节点的所有祖先节点中，找出所有“已经定位”的元素，其中距离该节点最近的一个节点，父亲比祖父近，祖父比曾祖父近，以此类推，“最近”的就是要找的定位基准。
4. 所谓“**已经定位**”元素的含义是，position属性被设置，并且被设置为不是static的任意一种方式，那么该元素就被定义为“已经定位”的元素。