**[CSS核心：包含块（Containing Block）](http://www.ddcat.net/blog/?p=1336" \o "点击阅读《CSS核心：包含块（Containing Block）》全文。)**

包含块（Containing Block）是视觉格式化模型的一个重要概念，它与框模型类似，也可以理解为一个矩形，而这个矩形的作用是为它里面包含的元素提供一个参考，元素的尺寸和位置的计算往往是由该元素所在的包含块决定的。

在阅读本文前，请先理解一下：

* [文档结构](http://www.ddcat.net/blog/?p=216)
* [元素类型](http://www.ddcat.net/blog/?p=835)
* CSS的[direction属性](http://www.w3.org/TR/CSS2/visuren.html" \l "propdef-direction" \o "W3C CSS2手册关于direction属性的说明)

**1视口（viewport）**

浏览器的窗口一般由3个部分组成，如图5所示。

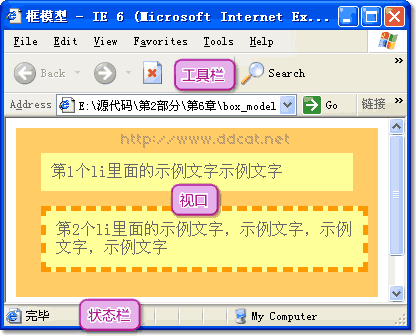
[](http://www.ddcat.net/web/css/boxmodel/005.png)

图5 浏览器的视口

连续媒介的用户端（例如电脑的浏览器）通常提供给用户一个视口（屏幕上的一个窗口或浏览区域），用户通过它来浏览文档。当视口尺寸改变时（例如调整浏览器的窗口大小），用户端可能会改变文档的布局。

如果视口比文档设定的大小要小，用户端往往会提供滚动机制（例如浏览器的滚动条）。

对于一个渲染区域而言，最多只能有一个视口，不过用户端可以对多个渲染区域加以渲染（即对同一文档提供不同的视口）。

**2 包含块**

在浏览器生成显示的页面的时候，每一个框都有一个定位，这个定位受其包含块的影响，不过它不被包含块所限制，而且可能会溢出到包含块之外。

CSS手册中，对于一个元素的包含块的定义如下：

1. 根元素存在的包含块称为**初始包含块**。
   * 在(X)HTML中，根元素是html元素（尽管有的浏览器会不正确地使用body元素）。
   * 而初始包含块的direction属性与根元素相同。(direction属性指定了块的基本书写方向，它还规定了表格列布局的方向、水平溢出的方向等。)
   * 初始包含块的宽度可以由根元素的width属性指定。如果该属性取值为“auto”，用户端提供初始宽度（如，视口的当前宽度）。
   * 初始包含块的高度可以由根元素的height属性指定。如果该属性取值为“auto”，包含块的高度将调整以**适应文档内容**（即由其内容决定其高度）。
   * 初始包含块不可以被定位或浮动（即，用户端忽略根元素的position和float属性）。
2. 对于其它元素：如果该元素的定位（position）为“relative（相对）”或者“static（静态）”，它的包含块由它最近的块级、单元格（table cell）或者行内块（inline-block）祖先元素的内容框创建。
3. 如果元素设定了“固定定位（position: fixed）”，包含块由视口创建。
4. 如果元素设定了“绝对定位（position: absolute）”，包含块由**最近的**position属性为“absolute”、“relative”或者“fixed”的祖先元素创建，方法如下：
   * 如果祖先元素是行内元素，包含块取决于祖先元素的direction属性：
     + 如果direction为“ltr”，包含块的顶、左边是该祖先元素创建的第一个框的顶、左补白边，它的底、右边是该祖先元素创建的最后一个框的底、右补白边。
     + 如果direction为“rtl”，包含块的顶、右边是该祖先元素创建的第一个框的顶、右补白边，它的底、左边是该祖先元素创建的最后一个框的底、左补白边。
   * 否则，祖先的补白边形成包含块。
   * 如果不存在这样的祖先元素，则元素的包含块为初始包含块。

上面的定义太抽象，所以我整理了一个流程图来说明一下，如图6所示。

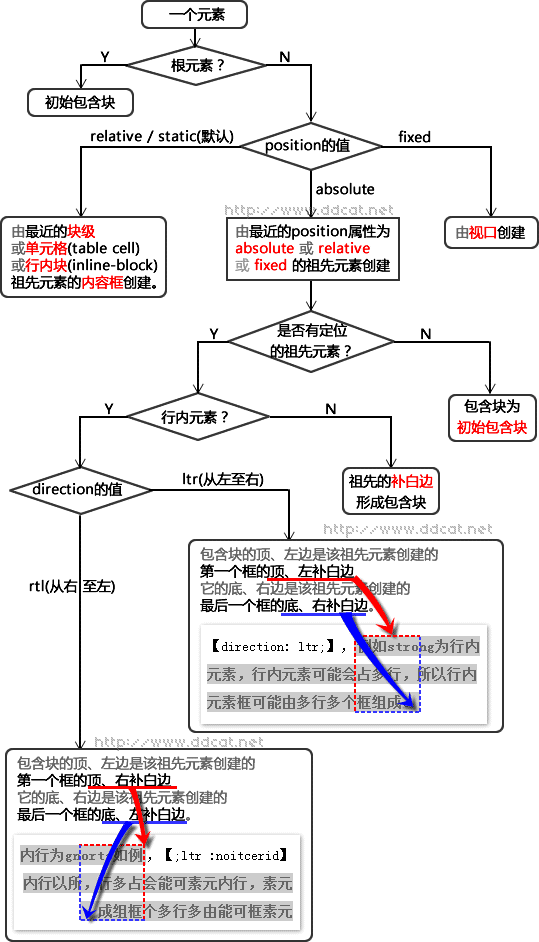
[](http://www.ddcat.net/web/css/boxmodel/006.png)

图5 浏览器的视口

**提示：**此时特别需要注意的是，祖先元素是行内元素时，浏览器会出现不一致的解释方式，大家可以在不同的浏览器内[测试一下](http://www.ddcat.net/web/css/boxmodel/direction.html)。

例如下面的XHTML文档，其文档结构及在浏览器内的显示如图7所示。当元素都没有定位的时候，包含块的创建如表1所示。

<!DOCTYPE HTML>

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="zh-CN" lang="zh-CN">

<head>

<title>2 包含块 :: 框模型 :: 豆豆猫的窝</title>

</head>

<body id="body">

<div id="div1">【div1】的文字。

<p id="p1">这里是【p1】。</p>

<p id="p2">这里是【p2】<em id="em1">【em1】<strong id="strong1">【strong1】</strong></em>。</p>

</div>

</body>

</html>

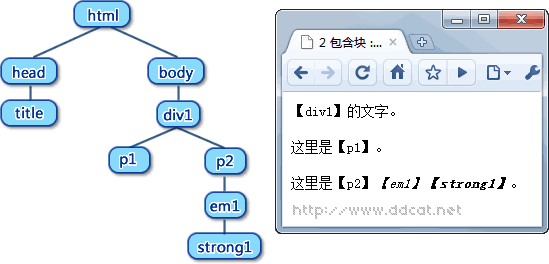
[](http://www.ddcat.net/web/css/boxmodel/007.png)

图7 文档结构图

|  |  |
| --- | --- |
| 表1 无定位元素的包含块的创建 | |
| **产生框的元素** | **为框创建包含块的元素** |
| body | 初始包含块（与用户端相关） |
| div1 | body |
| p1 | div1 |
| p2 | div1 |
| em1 | p2 |
| strong1 | p2 |

此时，如果对层“div1”设定定位：

#div1 {position:absolute; top:30px; left:20px; background:#0C6;}

则“div1”层的包含块不再是“body”，而是初始包含块（因为没有其它定位祖先框）。

如果再增加对“em1”的定位：

#em1 {position:absolute; top: 60px; left:50px; background:#FC3;}

则包含块的创建如表2所示。

|  |  |
| --- | --- |
| 表2 元素设定了定位的包含块的创建 | |
| **产生框的元素** | **为框创建包含块的元素** |
| body | 初始包含块 |
| div1 | 初始包含块 |
| p1 | div1 |
| p2 | div1 |
| em1 | div1 |
| strong1 | em1 |

“em1”定位后，它的包含块变为由它最靠近的祖先定位框“div1”创建的那个框，如图8所示。

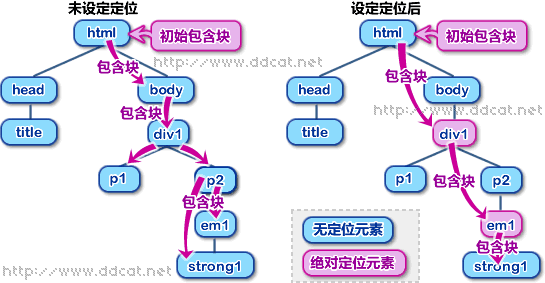
[](http://www.ddcat.net/web/css/boxmodel/008.png)

图8 改变元素定位属性对包含块的影响

因此对元素的左（left）和上（top）的位置的计算，如图9所示。（[查看实例文件](http://www.ddcat.net/web/css/boxmodel/containing_block.html)）

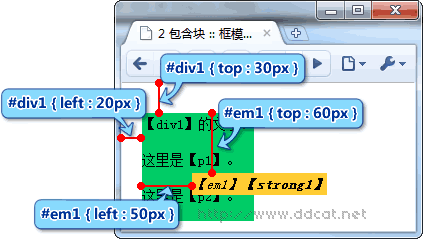
[](http://www.ddcat.net/web/css/boxmodel/009.png)

图9 定位元素位置的计算