
表格简述

表格是在制作网页的重要武器。诸如让自己的页面有底色、底图之类的修饰，只能通过表格来实现。表格是一个可自定义的容器，因而它就是我们用以实现页面构设的利器。

把表格称作容器缘于表格的装载特性。表格是用于装载数据的，与我们现实中所看到和理解的表格非常一致。表格不仅仅只能装载数据，它还被广泛地应用于网页的布局，这与表格可以将其所在区域划分为行和列即诸多单元格密切相关。页面布局将因使用了表格而可以随心所欲地排版。

在 HTML 里，表格有表头、表体、单元格、行、列等等概念，通过学习、领会和实践，是能够逐渐掌握的。

◆表格标签基本代码构成表达式

以下是一个完整的表格标签代码框架，为了能够一目了然，各元素尚未引入它们的属性：

```
<table>
  <tbody>
    <tr>
      <td>内容</td>
    </tr>
  </tbody>
</table>
```

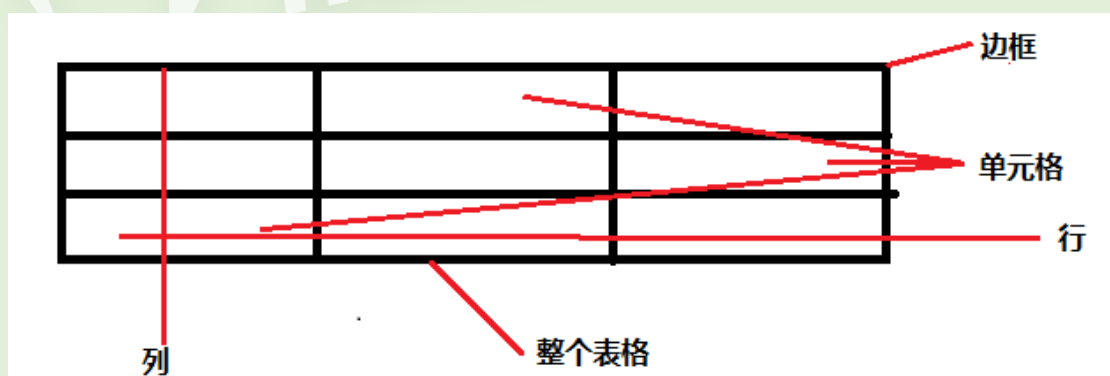
◆认识基本表格

一个表由 `<table>` 开始，`</table>` 结束，表的内容由 `<tr>`，`<th>` 和 `<td>` 定义。`<tr>` 说明表的一个行，表有多少行就有多

少个<tr> ; <th>说明表的列数和相应栏目的名称,有多少个栏就有多少个<th> ; <td>则填充由<tr>和<th>组成的表格。

表格重要的基本标记不多,但每个标记都有很多的属性,所以还是从表格的外观(即浏览器中显示的表格的样式)来了解最基本的属性。

先来看看表格的基本构造。下图是一个3行3列的表格：



这里面有两个概念要弄明白：**表格**与**单元格**。他们的关系是整体与局部的关系，如同砌墙的砖和砌好的墙一样。在上面3行3列的表格中一共有9个单元格。由于表格属性和单元格属性有很多是相同的，所以一定要分清楚。

表格的最基本标记为<table>、<tr>、<td>，可以先树立这样一个概念：描述整个表格的属性标记放在<table>里，描述单元格的属性标记放在<tr>、<td>里。

◆表格标签基本代码解析：

<table>：表格的起始符。任意一个表格的开始都必须以它开头，且必须有终止符</table>。

<tbody>：表体的起始符。紧跟在<table>之后，表示表体开始。必须有结尾符，放在</table>之前。避免导致代码出错。

`<tr>` :tr 的作用是规定表格的行 ,其中 ,t 是 table ,r 是 row(行)。有多少组 tr , 这张表格就有多少行。 `<tr>` 紧跟在 `<tbody>` 之后。必须有终止符 `</tr>`。

`<td>` :td 的作用是规定表格的列 ,t 是 table ,d 可理解为 down (向下)。有多少组 td , 这张表格就有多少列。第一个 `<td>` 紧跟在 `<tr>` 之后。终止符为 `</td>`。td 与 tr 配合构成单元格。

`</td>`、`</tr>`、`</tbody>`、`</table>` : 这些都是相应元素的终止符 , 表示相应元素所规范的内容结束。必须注意它们的排列顺序 , 不能有错。

从表格的代码框架分析得出 : 表格标签里 , 所有元素都成对出现 , 而这些元素是按照从大到小的顺序从外往里层层包裹的。table 最大 , 它占在起始和终结的位置 , tbody 是老二 , tr 是老三 , td 最小 , 它在最里层。

◆表格标签常用属性与参数设置 :

1、border : 表格边框线宽度。用于指定在边框外围添加的那些不同样式的线段参数。

例如 : border=1 , 表示表格边框的粗细为 1 个像素 (默认值) , 参数为 0 时表示没有边框线。

2、cellspacing : 单元格间距。用于指定单元格之间的距离参数 , 当一个表格有多个单元格时 , 各单元格的距离就是 cellspacing 指定的参数值了 , 如若表格只有一个单元格 , 那么 , 这个单元格与表格上、下、左、右边边框的距离也是 cellspacing。

例如 : cellSpacing=10 ; 这里的数字 10 , 是指 10px。

3、cellpadding：单元格边距，也可叫做衬距。用于指定该单元格里的内容与 cellpadding 区域的距离参数。

例如：cellpadding=5；这里的数字 5，是指 5px。则表示单元格里的内容与表格周边边框的距离。

4、width：表格的宽度。取值从 0 开始，默认以像素为单位，与显示器的分辨率的像素是一致的。例如：width="650"。

在设置表格的宽度时要充分考虑显示分辨率问题，如果表格设置的像素宽度太大，那么，得出的效果将导致 IE 的横向滚动条出现，只有通过滑动它才能看到表格最右边的内容。width 的取值还可以使用百分比，例如：width="100%"，这种赋值法有个好处：表格的宽度将根据可显示的宽度来自我调整宽度。

5、height：表格的高度，取值方法同 width。提示：如果不是特别需要，不用设置表格的高度，系统会根据表格的内容自动设置高度。所谓特别的需要，是指一些特殊的情形下，需要表格的高度精确，比如，当我们通过表格的背景来发一张图片时，如果表格的高度不精确定义，图片就不可能完整或完美地显示。

6、bgcolor：表格的背景色。取值方法举例：bgcolor=#ff0000 或 bgcolor=red。单元格<td>也可有此属性，如果设置了表格的背景色，又设置表格单元格的背景色，得出的效果将是值得试试的，这种情况主要用于多单元格的表格。

7、background：表格的背景图。其值为一个有效的图片地址。<td>也有此属性。

8、bordercolor：表格的边框颜色，当 border 值不为 0 时设置此值有效。取值同 bgcolor。

9、bordercolorlight : 亮边框颜色，当 border 值不为 0 时设置此值有效。亮边框指表格的左边和上边的边框。

10、bordercolordark : 暗边框颜色，当 border 值不为 0 时设置有效。暗边框指表格的右边和下边的边框。

11、border-style : 边框线样式，表达方式为：
style="border-style: groove;"

12、align : 表格的对齐方式，值有 left (左对齐，默认) center (居中) 以及 right (右对齐)。

align 定义的是表格自身的位置，这是一个很有用的属性，强烈建议使用它来规定表格的对齐方式，尽量不要使用 <p align=?>表格</p>、<div align=?>表格</div>和<center>表格</center>之类的标签来规定表格的位置，因为这么做将导致添加许多废码！此外，当表格的宽度设置为 100%，或者，表格的宽度设置成了占满它所在的容器的宽度，没有必要定义 align 属性。

★单元格属性设置

从表格的整体外观上看，单元格是表格的基本构成单元，一张表格至少要有有一个单元格。单元格由<tr><td></td></tr>共同组成，但在的一组<tr>里，可以拥有多组<td>，有多少组 td 该表格就有多少列；同样，一张表格可以拥有多组 tr，表格有多少组 tr 就有多少行。显而易见，构成表格最小单元的 td，它的属性设置是不可忽视的。

单元格的属性设置主要体现在<td>里，除了在上面学到的 bgcolor 等之外，还有许多属性需要掌握，令人高兴的是，很多属性

的设置方法都是和<table>属性设置一样的，如 width、height、bordercolor 等。

对于 width，在运用时要特别注意。好的习惯是，如果是一个单元格的表格，假设在<table>里设置了其宽度，那么，就不必再设置<td>的宽度了，因为这是多余的，如果再另行这么设置：<td width="100%">，显然只是增加了代码量，它没有任何作用——当有了表格宽度，单元格的宽度会自动占满表格的宽度。但应引起重视的是：在多列表格里，<td>的宽度就应该设置，如若不设置，系统会自动按各单元格里内容的占位情况自行分配宽度，效果是可想而知的。

单元格的 bordercolor 设置有些讲究，主要用于实现单像素的表格边框——要实现这种效果，要求表格的 border 为 1，且<table>里设置了 bordercolor 颜色值，cellpadding 和 cellspacing 都设置为 0，然后将<td>的 bordercolor 颜色值设置为与<td>的背景色相一致。

单元格的最为重要的属性应该为 align 和 valign，align 规范单元格里内容的水平对齐方式，valign 则规定垂直方向的对齐方式——请特别注意：它们规定的对齐方式是单元格里面的内容，而不象表格里的 align 那样定义自身的位置！

align 取值方法和前面介绍的一样，valign 的取值方法主要有 left（左对齐）、middle（居中）、top（靠顶）、bottom（靠底）等。

★表格组合

表格组合更多的用于布局，需要熟练运用各种技巧，掌握它具有一定的挑战性，但表格组合技巧是一个很有趣的技术。要顺利地实现 N 张表格的纵向组合，至少有三种方法：

一是，令参与排列的表格都居中（即在<table>里加入 align=center ）。

二是，不定义表格的对齐属性，即<table>里没有 align 属性，这样给出的 N 张表格代码，它们都是上下排列的，表格的自身位置全是左对齐（但若定义了 align=left，情形不是这样，可以在 DW 里试试）。

三是把参与排列的表格放在一张表格里，这张作为容器的外表宽度不限，且里面的表格也不设置 align 属性——大家都看得出来，其实它只是第二种实现方法的扩展，其目的是为了让参与排列的表格随心所欲地设置水平对齐。换句话说，外表的作用是为了实现自定义的对齐方式，第三种方法虽然源于第二种，但使用它是个绝好的主意，这个，在使用表格排版过程中，相信会慢慢体会到的。

水平排列的表格组合

表格的水平排列更需要精确地计算参与排列的表格的宽度，它们的宽度的总和不能超过其所属容器所能装载的宽度，表格在和其他元素的混排情形下更应该严格考虑。

一个好的习惯是：用一张表格来规范并列组合的表格，这张表格，最好宽度设置为 100%，然后将里面要并列组合的表格全部设置为左对齐（但注意：它们的宽度总和不能超过可显示的宽度）。

这里从某个角度分析了表格的组合问题，它是一个主要用于排版的技术，要求计算精确，并不十分适用于论坛。但如果你是一个不怕挑战的人，那么，这里所探讨的内容或许对你来说还远远不够，还需要发挥聪明才智去尝试。

★多单元格表格

在页面布局方面，比起表格组合，多单元格更能灵活和方便地进行控制，使用多单元格布局是一个不错的习惯。

前面提过，表格代码里，有多少组<tr>...</tr>，表格就有多少行，有多少组<td>...</td>，表格就有多少列。因此，在代码里，要根据需要利用表格代码里的<tr><td>...</td></tr>来创建单元格。

例：制作一个 1×3 即一行三列的表格。

代码：

```
<table border=1 cellpadding=0 cellspacing=0 width=450
align=center>
  <tbody>
    <tr>
      <td align=center>单元格一（即列 1）</td>
      <td align=center>单元格二（即列 2）</td>
      <td align=center>单元格三（即列 3）</td>
    </tr>
  </tbody>
</table>
```

效果：(1×3 表格)

单元格一（即列1）	单元格二（即列2）	单元格三（即列3）
-----------	-----------	-----------

在上述示例里，只有一组<tr>，故而表格只有一行，<tr>...</tr>里有三组 td，因此，表格有三列，这就是所谓的 1×3 表格。现在我们把代码扩展一下，即增加一组 tr，请看代码和效果：

代码：

```
<table border=1 cellpadding=0 cellspacing=0 width=450
```

align=center>

<tbody>

<tr>

<td align=center>单元格一（即列 1）</td>

<td align=center>单元格二（即列 2）</td>

<td align=center>单元格三（即列 3）</td>

</tr>

<tr>

<td align=center>单元格四（即列 4）</td>

<td align=center>单元格五（即列 5）</td>

<td align=center>单元格六（即列 6）</td>

</tr>

</tbody>

</table>

效果：(2×3 表格)

单元格一（即列1）	单元格二（即列2）	单元格三（即列3）
单元格四（即列4）	单元格五（即列5）	单元格六（即列6）

我们可以定义<td>里的宽度和高度值，从而让表格以我们的预设在其所在区域进行布局，以便达到有效的排版目的。

在多行多列的表格里，当某一列的第一个单元格已经定义了宽度，那么，往下各单元格的宽度将跟它一致，如果同一列的所有单元格都设置了宽度，该列的最终宽度以 width 值最大的那个单元格为标准，——由此可见 我们只需设置某列的第一个单元格的宽度即可，其他同列的<td>不必给出 width 属性，这样一可以节省代码二不会出现无效设置。

表格的行的高度情形也是一样：当定义了某一行头一个单元格的高度之后，该行其他单元格会自动得到相同的高度值，不必再一一设置<td>的 height 属性。尽量减少代码量总是一个好习惯，一来可以节省空间，二来打开速度也会得到改善——代码越多，系统处理代码的时间就越长，同时也更耗可贵的系统资源。

我们还应该注意的是，如果不设置<td>的 width 和 height 属性，表格各单元格将根据里面的内容的占位情况拓宽自己，表格给各单元格的宽度分配将由系统自动进行，这样就容易出现不合理的情况，因此，应该根据需要设置单元格的高度和宽度。

多单元格表格里一个有趣的问题是单元格合并，其概念与 Excel 的合并单元格是一致的，只是实现方法不同。在 HTML 代码里，我们用 colspan 属性合并左右单元格，记作：colspan=数值，“数值”即为需要合并的单元格总数；而 rowspan 属性则合并上下单元格，记作：rowspan=数值。

让我们用上面的第二组示例表格来加工一下。先看**合并左右的单元格**：

代码：

```
<table border=1 cellpadding=0 cellspacing=0 width=450  
align=center>  
<tbody>  
<tr>  
  <td align=center colspan=2>1，2 单元格合并了</td>  
  <td align=center>3</td>  
</tr>  
<tr>  
  <td align=center>4</td>
```

```
<td align=center>5</td>
<td align=center>6</td>
</tr>
</tbody>
</table>
```

效果：

1, 2单元格合并了		3
4	5	6

下面是表格的**上下单元格合并**：

代码：

```
<table border=1 cellpadding=0 cellspacing=0 width=450
align=center>
<tbody>
<tr>
<td align=center rowspan=2>1,4 单元格合并了</td>
<td align=center>2</td>
<td align=center>3</td>
</tr>
<tr>
<td align=center>5</td>
<td align=center>6</td>
</tr>
</tbody>
</table>
```

效果：

1,4单元格合并了		2	3
		5	6

通过 HTML 代码实现单元格的合并略显抽象一些，还可以研究本示例代码和第二组代码(2×3 表格)的区别并比较两组表格的样式来慢慢领会，要完全掌握这一技术，还需要亲自做些尝试。

★表格嵌套

作为容器，表格可以往里放置可以放置的元素。表格也是一个元素，故而表格放在表格里是合理的，因此就有了表格的嵌套。

表格的嵌套还可以是另外的形式，那就是，一个表格的单元格里还有表格。不论是哪种形式的嵌套，理论上嵌套的层可以是无穷尽的，但现实中并不会这么做，我们总是根据需要有限制地使用嵌套，不会没完没了。

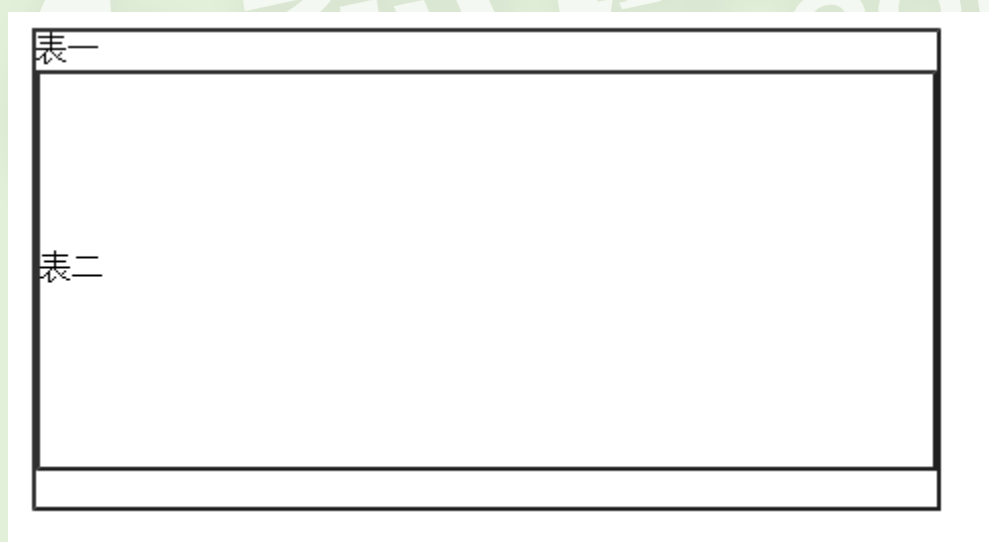
表格嵌套其实也不复杂。如果你能够独立用代码做一张表格(强烈建议练出这样的本事)，那么，你就可以做一个两张表格的嵌套，很简单，把代码复制一下，粘贴在原代码的<td>和</td>之间(亦即在“和”的位置)。

试看以下代码和两个表格的嵌套效果：

```
<table border=1 cellpadding=0 cellspacing=0 width=450
height=200 align=center> <tbody>
<tr>
<td>表一
<table border=1 cellpadding=0 cellspacing=0 width=450
height=200 align=center> <tbody>
<tr>
<td>表二</td>
```



```
</tr>
</tbody> </table>
</td>
</tr>
</tbody> </table>
```



从上面的表格效果上看，如果我们把文字“表一”删除掉，表格的嵌套将是非常完好的。

如果再定义边框大小和颜色、cellpadding 和 cellspaing 为其他值，加上背景色和背景图，或通过其他手段来修饰，那么，里表和外表之间，就会出现非常漂亮的边框，日志的外框就这么搞掂了。

当然，这是一个细活，需要做反复的尝试，直到效果满意为止。或许你还需要更多层的嵌套，这就看你的想象力了——实现手段是不难的，因为你已经会了第一层的嵌套，第二层、第三层……第 N 层，无非就是如此。

现在我们来看看多单元格表格中，在单元格是嵌入表格的示例。先看代码：

```
<table border=1 cellpadding=0 cellspacing=0 width=450
```

```
align=center> <tbody>
<tr>
  <td class=red>单元格一：无表格嵌套</td>
  <td>
    <table border=1 bgcolor=#ccc1cc width="100%"> <tbody>
      <tr> <td class=red>我嵌套在单元格二里</td>
      </tr> </tbody> </table>
    </td>
  </tr>
</tbody> </table>
```

以下是效果：

单元格一：无表格嵌套	我嵌套在单元格二里
------------	-----------

上面的两个例子里，都是将处于外层的表格（即外表）的宽度设置为 450 个像素。在实际制作中，要充分考虑显示器的分辨率可显示宽度来决定取值。

好的习惯仍然是取绝对值，因为，日志里边的内容宽度非常可能是绝对的，譬如图片有着自己固定的宽度，Flash 影片也是使用绝对的 width 值，为了不让正文四周出现太多的空余，采用绝对值规范外表的宽度是有必要的。

当然，还有一个好的做法，那就是，根本不用规定外表的宽度，而是让里边的内容自行撑宽外表，这么做至少要求我们注意两个方法的问题：

一是利用 cellpadding 或 border 的合理值来让外表的边缘显示

外表的既设背景。

二是外表里面的诸元素的宽度要相对统一。任何一种实现方式都有其自身的奥妙，不是用语言能够一一道明的，只有在不断地实践中才能逐一领悟和掌握。

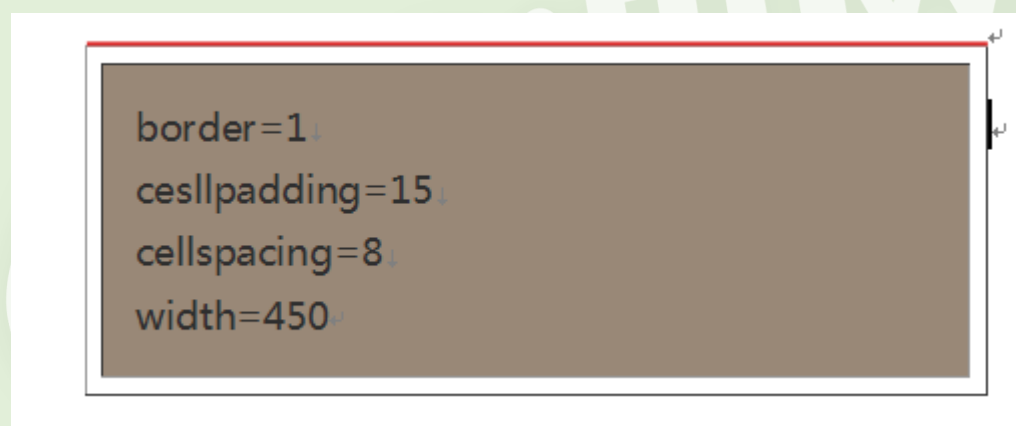
当今用户的的显示器分辨率有两大标准：800×600 和 1024×768，这两种分辨率下，可显示的区域宽度是有区别的。为此，需要通过尝试获知具体宽度，从而制作出的帖子不至于出现浏览者不能完整欣赏的现象。过宽的像素的宽度，分辨率设置为 800×600 的浏览者将看不到右边的内容，而太小的宽度，又可能显得小里小气，也不理想。

★深入学习表格

表格还有一些重要的内容需要彻底弄懂。实际上，这些内容，前面已经接触过，但还需要进一步去探讨一下。

实验一：

下面给出一条长度为 450 个像素的分隔线，然后在分隔线下面放置一张表格，其宽度亦为 450 个像素。

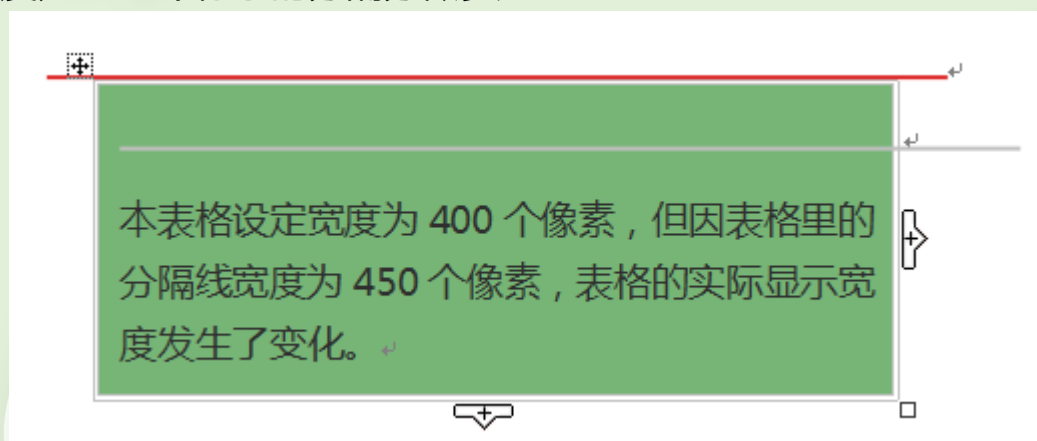


现在，我们看到，表格的总宽度仍然是 450 个像素（它没有超

越分隔线)，因此，我们可以得出这样的结论：表格的 border、cellpadding 和 cellspacing 虽然都是占位元素，但它们不会使得表格变得更宽，换一个更生动的说法，表格的这些元素是往里占位的，而不是往外占位。

实验二：

设定表格的宽度为 400 个像素，然后在里面放置一个可视元素，令其宽度为 450，看看表格的总宽度发生了什么。我们依然用一条长度为 450 个像素的分隔线做参照。

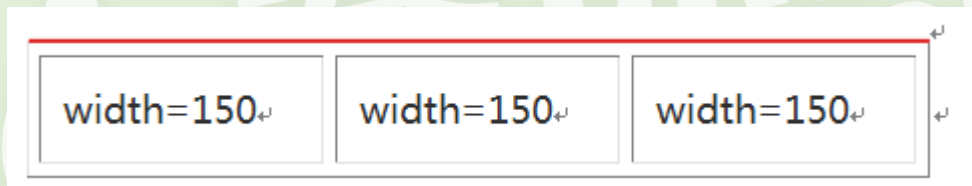


现在我们可以看到，虽然在 `<table>` 标签里设定了 `width=400`，但表格的总宽度并不是 400 个像素，而是被表格里面的那条分隔线的实际宽度（450 个像素）撑宽了。你可能会觉得奇怪：为什么表格总宽度超过了 450 个像素呢？道理是这样的：表格设置了 cellpadding 值为 10，cellpadding 区域是不可放进任何可视内容的。细心比较一下，将会发现，表格宽出的部分正是留空的部分，即表格的单元格衬距。

可见，表格里面的内容会影响到表格的宽度。实际上，如果我们设置好表格里面的内容的宽度，表格的 width 值是完全可以不设置的，但这不是一个好的习惯，我们通常仍然需要合理地设置表格的宽度，但必须留意一个问题：表格的总体宽度是否超过了可显示的范围。

实验三：

设定一个 450 个像素宽度的表格，表格的边框、单元格间距和衬距都为实值，该表格有三个单元格，每个单元格设定宽度为 150（加起来恰好的 450 个像素），看看表格的实际宽度是否发生变化。这个实验仍然使用 450 个像素宽度的分隔线作参照。



上面的表格，我们设置 `border=1`，`cellpadding=10`，`cellspacing=6`，三个单元格的宽度均为 150 个像素，结果是：表格的总体宽度未发生变化。由此我们得出结论：多单元格表格里，只要各单元格的宽度相加起来不超过 `<table>` 里的宽度赋值，它们不会因表格边框、单元格间距和单元格衬距都为实值而撑宽表格。

通过实验的方式，从表格的宽度角度剖析了表格的一些深层特征，它并不是很抽象，利用分隔线做对照标准和多做实验，会很容易掌握的。

★表格应用示例





这是用一个表格嵌套的示例来展示制作过程。以下是本示例的参考代码:

```
<html>
<head>
<title>超级链接的设置</title>
<meta charset="utf-8">
</head>
<body>
<table border=1 bordercolor=#363636 cellpadding=20
```

```
cellspacing=0 bgcolor=#dd6969 align=center
width=680> <tr> <td bordercolor=#ff0000>
    <table border=1 bordercolor=#696969 cellpadding=10
cellspacing=0 bgcolor=#aa1212 width="100%"> <tr> <td
bordercolor=#aa1212>
    <table border=0 cellpadding=10 cellspacing=0
bgcolor=#ffffff width="100%"> <tr> <td>
    <table border=0 cellpadding=10 cellspacing=0
bgcolor=#dd6969 width="100%" height=40
align=center> <tr> <td>
<p align=center> <font face=黑体 size=4 color=#ffcc00>标
题</font> </p>
    </td> </tr> </table>
    <table border=0 cellpadding=0 cellspacing=0
width="100%" height=10
bgcolor=#ffffff> <tr> <td> </td> </tr> </table>
    <table border=0 cellpadding=0 cellspacing=10
bgcolor=#393939 width="100%" height=450
align=center> <tr> <td valign=top>
<p> <font color=#ffffff>内容</font> </p>
    </td> </tr> </table>
    </td> </tr> </table>
    </td> </tr> </table>
</td> </tr> </table>
</body>
</html>
```

本例有四层表格的嵌套，总共使用了六张表格。最里层是三张表格上下排列，属表格组合。我们从外往里分析代码。

最外层的表格，我们设置边框值为 1 并定义其颜色，这样，配合 <td> 的边框颜色能够起到外边缘呈现出立体的感觉。外表的单元格衬距 (cellpadding) 为 20，它恰好形成了日志外边框。外表的宽度为 680 个像素，这是根据本页面而设置的，可以在实际使用时根据需要更改。

第二层表格的原理和外层的差不多，只是更改了各元素的颜色，单元格衬距也调小了，cellpadding 赋值为 10。

第三层表格没有做太多的设置，仅令其背景色为白色、单元格衬距为 10，这样得出一个白色的边框。

第四层表格有三个上下并列组合的表格。第一个用来日志的标题；第二个表格只是用于隔开第一个和第三个表格并令之与第三层白色底的表格连为一体，为此它的 border、cellpadding 和 cellspacing 都需要设置为 0，高度则为 10（与第三层表格所形成的边框一致）；第三个表格用于放置帖子的内容，故将其单元格间距设置为 10（也可设置单元格衬距），以便让文字等内容不至于太靠边。

