

【纯转】Div+CSS 经典速成教程。

2010-01-12 14:46

今天看到一个 Div+CSS 布局的经典教程，奉献给大家，本人一向偏好 DIV+CSS，但由于关注 seo 和网站运营较多，造成了 DIV+CSS 做网页模板的技术生疏，为了培养自己成全才，不得不好好消化一翻，其实很简单滴。这个 DIV+CSS 布局教程很不错。

教程共分 5 节，其实就一点点内容，不过通过学习，一定能让你领会到 DIV+CSS 的精髓，不要嫌文字多，一个小时一定能让你学完，当然是针对熟悉网页制作基础的朋友。

在网页制作中，有许多的术语，例如：CSS、HTML、DHTML、XHTML 等等。在下面的文章中我们将会用到一些有关于 HTML 的基本知识，而在你学习这篇入门教程之前，请确定你已经具有了一定的 HTML 基础。下面我们就开始一步一步使用 DIV+CSS 进行网页布局设计吧。

所有的设计第一步就是构思，构思好了，一般来说还需要用 PhotoShop 或 FireWorks (以下简称 PS 或 FW) 等图片处理软件将需要制作的界面布局简单的构画出来，以下是我构思好的界面布局图。

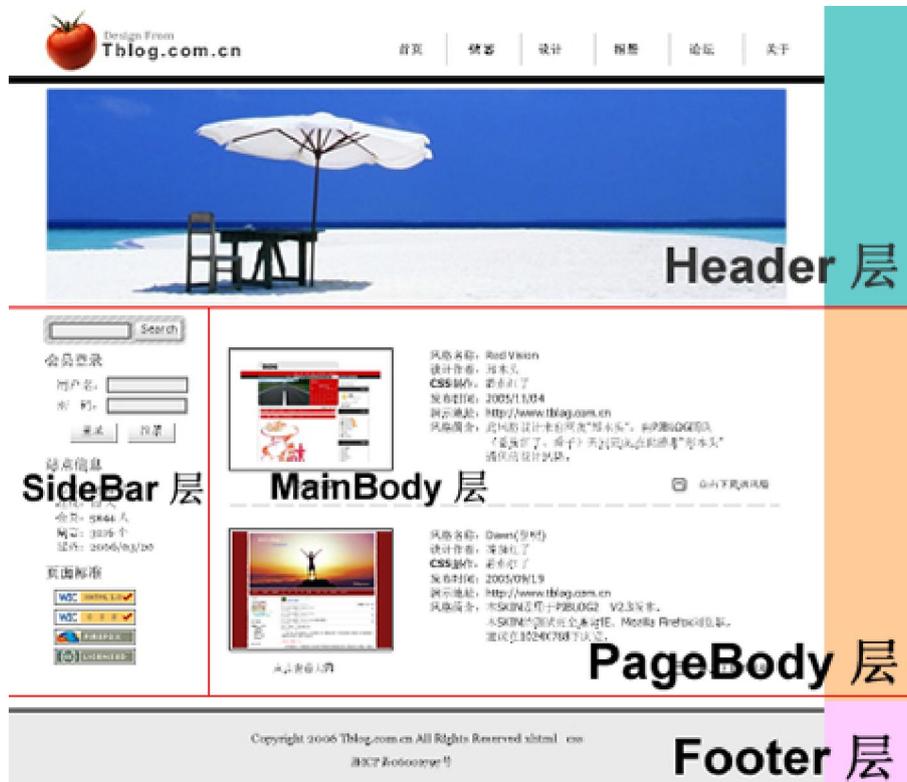


下面，我们需要根据构思图来规划一下页面的布局，仔细分析一下该图，我们

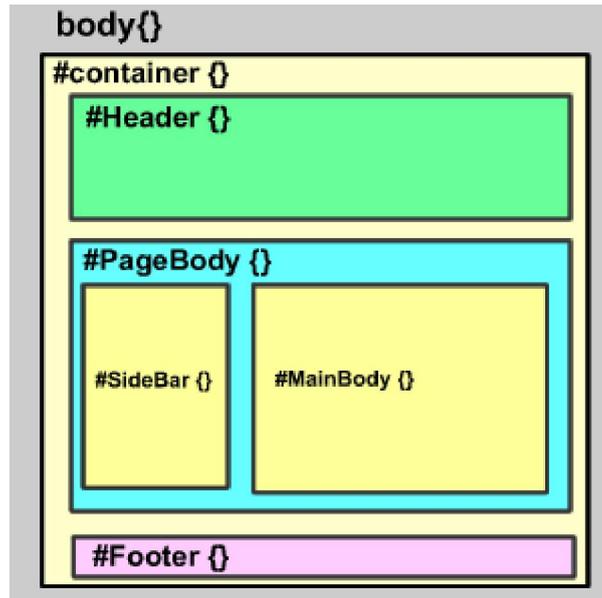
不难发现，图片大致分为以下几个部分：

- 1、顶部部分，其中又包括了 LOGO、MENU 和一幅 Banner 图片；
- 2、内容部分又可分为侧边栏、主体内容；
- 3、底部，包括一些版权信息。

有了以上的分析，我们就可以很容易的布局了，我们设计层如下图：



根据上图，我再画了一个实际的页面布局图，说明一下层的嵌套关系，这样理解起来就会更简单了。



DIV 结构如下:

```
|body {} /*这是一个 HTML 元素, 具体我就不说明了*/
└─#Container {} /*页面层容器*/
    └─#Header {} /*页面头部*/
    └─#PageBody {} /*页面主体*/
        └─#Sidebar {} /*侧边栏*/
        └─#MainBody {} /*主体内容*/
    └─#Footer {} /*页面底部*/
```

至此, 页面布局与规划已经完成, 接下来我们要做的就是开始书写 HTML 代码和 CSS。

接下来我们在桌面新建一个文件夹, 命名为“DIV+CSS 布局练习”, 在文件夹下新建两个空的记事本文档, 输入以下内容:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>无标题文档</title>
<link href="css.css" rel="stylesheet" type="text/css" />
</head>

<body>
</body>
</html>
```

这是 XHTML 的基本结构，将其命名为 index.htm，另一个记事本文档则命名为 css.css。

下面，我们在<body></body>标签对中写入 DIV 的基本结构，代码如下：

```
<div id="container">[color=#aaaaaa]<!--页面层容器-->[/color]
  <div id="Header">[color=#aaaaaa]<!--页面头部-->[/color]
  </div>
  <div id="PageBody">[color=#aaaaaa]<!--页面主体-->[/color]
    <div id="Sidebar">[color=#aaaaaa]<!--侧边栏-->[/color]
    </div>
    <div id="MainBody">[color=#aaaaaa]<!--主体内容-->[/color]
    </div>
  </div>
  <div id="Footer">[color=#aaaaaa]<!--页面底部-->[/color]
  </div>
</div>
```

为了使以后阅读代码更简易，我们应该添加相关注释，接下来打开 css.css 文件，写入 CSS 信息，代码如下：

```
/*基本信息*/
body {font:12px Tahoma;margin:0px;text-align:center;background:#FFF;}

/*页面层容器*/
#container {width:100%}

/*页面头部*/
#Header {width:800px;margin:0 auto;height:100px;background:#FFCC99}

/*页面主体*/
#PageBody {width:800px;margin:0 auto;height:400px;background:#CCFF00}

/*页面底部*/
#Footer {width:800px;margin:0 auto;height:50px;background:#00FFFF}
```

把以上文件保存，用浏览器打开，这时我们已经可以看到基础结构了，这个就是页面的框架了。

关于以上 CSS 的说明（详细请参考 CSS2.0 中文手册，网上有下载）：

1、请养成良好的注释习惯，这是非常重要的；

2、body 是一个 HTML 元素，页面中所有的内容都应该写在这标签对之内，我就不多说了；

3、讲解一些常用的 CSS 代码的含义：

font:12px Tahoma;

这里使用了缩写，完整的代码应该是：font-size:12px;font-family:Tahoma; 说明字体为 12 像素大小，字体为 Tahoma 格式；

margin:0px;

也使用了缩写，完整的应该是：

margin-top:0px;margin-right:0px;margin-bottom:0px;margin-left:0px

或

margin:0px 0px 0px 0px

顺序是 上 / 右 / 下 / 左，你也可以书写为 margin:0(缩写)；

以上样式说明 body 部分对上右下左边距为 0 像素，如果使用 auto 则是自动调整边距，

另外还有以下几种写法：

margin:0px auto;

说明上下边距为 0px，左右为自动调整；

我们以后将使用到的 padding 属性和 margin 有许多相似之处，他们的参数是一样的，

只不过各自表示的含义不相同，margin 是外部距离，而 padding 则是内部距离。

text-align:center

文字对齐方式，可以设置为左、右、中，这里我将它设置为居中对齐。

background:#FFF

设置背景色为白色，这里颜色使用了缩写，完整的应该是 background:#FFFFFF。background 可以用来给指定的层填充背景色、背景图片，以后我们将用到如下格式：

background:#ccc url('bg.gif') top left no-repeat;

表示：使用#CCC(灰度色)填充整个层，使用 bg.gif 做为背景图片，

top left

表示图片位于当前层的左上端，no-repeat 表示仅显示图片大小而不填充满整个层。

top/right/left/bottom/center

用于定位背景图片，分别表示 上 / 右 / 下 / 左 / 中；还可以使用

background:url('bg.gif') 20px 100px;

表示 X 座标为 20 像素，Y 座标为 100 像素的精确定位；

repeat/no-repeat/repeat-x/repeat-y

分别表示 填充满整个层 / 不填充 / 沿 X 轴填充 / 沿 Y 轴填充。

height / width / color

分别表示高度(px)、宽度(px)、字体颜色(HTML 色系表)。

4、如何使页面居中？

大家将代码保存后可以看到，整个页面是居中显示的，那么究竟是什么原因使得页面居中显示呢？

是因为我们在#container 中使用了以下属性：

```
margin:0 auto;
```

按照前面的说明，可以知道，表示上下边距为 0，左右为自动，因此该层就会自动居中了。

如果要想页面居左，则取消掉 auto 值就可以了，因为默认就是居左显示的。

通过 *margin:auto* 我们就可以轻易地使层自动居中了。

5、这里我只介绍这些常用的 CSS 属性了，其他的请参看 CSS2.0 中文手册。

当我们写好了页面大致的 DIV 结构后，我们就可以开始细致地对每一个部分进行制作了。

在上一章中我们写入了一些样式，那些样式是为了预览结构而写入的，我们把 *css.css* 中的样式全部清除掉，重新写入以下样式代码：

样式说明：

```
a:link,a:visited {font-size:12px;text-decoration:none;}
```

```
a:hover {}
```

这两项分别是控制页面中超链接的样式，具体我就不说明了，请大家参阅手册。

```
#container {width:800px;margin:10px auto}
```

指定整个页面的显示区域。

width:800px 指定宽度为 800 像素，这里根据实际所需设定。

margin:10px auto，则是页面上、下边距为 10 个像素，并且居中显示。

上一章中我们讲过，对层的 *margin* 属性的左右边距设置为 *auto* 可以让层居中显示。

接下来，我们开始制作 TOP 部分，TOP 部分包括了 LOGO、菜单和 Banner，首先我们要做的就是对设计好的图片进行切片，以下是在 FW 下完成的切片：



我将 TOP 部分切片为两部分，第一部分包括了 LOGO 和一条横线。由于 LOGO 图片并没有太多的颜色，这里我于是将这一部分保存为 GIF 格式，调色板选择为精确，选择 Alpha 透明度，色版为白色(此处颜色应与背景色相同)，导出为 logo.gif，图像宽度为 800px。

到这里，有的朋友就说了，* 为什么要使用 GIF 格式？使用 JPEG 不是更好吗？因为 GIF 格式的图片文件更小，这样能使页面载入的速度更快，当然使用此格式之前必须确定图片并没有使用太多的颜色，当我们使用了 GIF 格式时，从肉眼上并不能看出图片有什么太大的变化，因此这是可行的。

* 接下来的 Banner 部分还能使用 GIF 格式吗？

答案是不能，因为 Banner 部分是一个细致的图片，如果使用 GIF 格式颜色会有太大的损失，所以必须使用 JPEG 格式，将文件导出为 banner.jpg。

*** 合理的切片是非常之重要的，因为切片的方法正确与否决定了 CSS 书写的简易程度以及页面载入速度。**

切好片后，我们还需要对 TOP 部分进行分析并将 DIV 结构写入 Header 中代码如下：

```
<div id="menu">
  <ul>
    <li><a href="#">首页</a></li>
    <li class="menuDiv"></li>
```

```
<li><a href="#">博客</a></li>
<li class="menuDiv"></li>
<li><a href="#">设计</a></li>
<li class="menuDiv"></li>
<li><a href="#">相册</a></li>
<li class="menuDiv"></li>
<li><a href="#">论坛</a></li>
<li class="menuDiv"></li>
<li><a href="#">关于</a></li>
</ul>
</div>
<div id="banner">
</div>
```

为什么要这么写呢，因为对菜单使用列表形式，可以在以后方便对菜单定制样式。

而为什么要添加以下代码呢？

```
<li class="menuDiv"></li>
```

插入这一段代码是可以方便地对菜单选项之间插入一些分隔样式，例如预览图中的竖线分隔。

然后我们在 css.css 中再写入以下样式：

```
/*页面头部*/
#header {background:url(logo.gif) no-repeat}
```

样式说明：

```
#header {background:url(logo.gif) no-repeat}
```

给页面头部分加入一个背景图片 LOGO，并且不作填充。

这里，我们没有指定 header 层的高度，为什么不指定呢？

因为 header 层中还有菜单和 banner 项，所以层的高度暂时是未知的，而层的属性又可以让层根据内容自动设定调整，因此我们并不需要指定高度。

使用列表制作菜单

开始此节的学习前，请确认你已经参照之前的几节内容写入了 DIV、CSS 到 index.htm 和 css.css 文件中。

这一节我将告诉大家如何用列表来制作菜单。

```
<div id="menu">
<ul>
  <li><a href="#">首页</a></li>
  <li class="menuDiv"></li>
  <li><a href="#">博客</a></li>
  <li class="menuDiv"></li>
  <li><a href="#">设计</a></li>
  <li class="menuDiv"></li>
  <li><a href="#">相册</a></li>
  <li class="menuDiv"></li>
  <li><a href="#">论坛</a></li>
  <li class="menuDiv"></li>
  <li><a href="#">关于</a></li>
</ul>
</div>
```

以上是这部分的结构，有关于、这两个 HTML 元素大家自己去参考相关的内容吧，它们最主要的作用就是在 HTML 中以列表的形式来显示一些信息。

还有一点需要大家一定要分清楚的，当在 HTML 中定义为 `id="divID"` 时，在 CSS 对应的设置语法规则是 `#divID {}`，如果在 HTML 中定义为 `class="divID"` 时，则在 CSS 中对应的设置语法是 `.divID`。

如果 `id="divID"` 这个层中包括了一个 ``，则这个 `img` 在 CSS 中对应的设置语法应该是 `#divID img {}`，同样，如果是包含在 `class="divID"` 这个层中时，则设置语法应该是 `.divID img {}`，这一点希望大家要分清楚了。

另外，HTML 中的一切元素都是可以定义的，例如 `table`、`tr`、`td`、`th`、`form`、`img`、`input`... 等等，如果你要在 CSS 中设置它们，则直接写入元素的名称加上一对大括号 `{}` 就可以了。所有的 CSS 代码都应该写在大括号 `{}` 中。

按照上面的介绍，我们先在 `css.css` 中写入以下代码：

```
#menu ul {list-style:none;margin:0px;}
#menu ul li {float:left;}
```

解释一下：

```
#menu ul {list-style:none;margin:0px;}
```

`list-style:none`，这一句是取消列表前点，因为我们不需要这些点。

`margin:0px`，这一句是删除 UL 的缩进，这样做可以使所有的列表内容都不缩进。

```
#menu ul li {float:left;}
```

这里的 float:left 的左右是让内容都在同一行显示，因此使用了浮动属性(float)。

到这一步，建议大家先保存预览一下效果，我们再添加下面的内容，效果如下：

首页 博客 设计 相册 论坛 关于

这时，列表内容是排列在一行，我们在#menu ul li {}再加入代码 margin:0 10px

```
#menu ul {list-style:none;margin:0px;}
```

```
#menu ul li {float:left;margin:0 10px}
```

margin:0 10px 的作用就是让列表内容之间产生一个 20 像素的距离(左: 10px, 右: 10px)，预览的效果如下：

首页 博客 设计 相册 论坛 关于

现在，雏形已经出来了，我们再来固定菜单的位置，把代码改成如下：

```
#menu {padding:20px 20px 0 0}
```

```
/*利用 padding:20px 20px 0 0 来固定菜单位置*/
```

```
#menu ul {float:right;list-style:none;margin:0px;}
```

```
/*添加了 float:right 使得菜单位于页面右侧*/
```

```
#menu ul li {float:left;margin:0 10px}
```

这时，位置已经确定了，可是构思图中，菜单选项之间还有一条竖线，怎么办呢？

别忘了，我们早就已经留好了一个空的<li class="menuDiv">，要想加入竖线就使用它了。

按照上面说的方法，我们再添加以下代码：

```
.menuDiv {width:1px;height:28px;background:#999}
```

保存预览一下，竖线是否已经出来了？关于这段代码就不多讲了，应该是很容易理解的。

首页 | 博客 | 设计 | 相册 | 论坛 | 关于

不过，菜单选项的文字却在顶部，我们再修改成以下代码：

```
#menu ul li {float:left;margin:0 10px;display:block;line-height:28px}
```

关于 **display:block;line-height:28px** 大家可以去参阅一下手册，我就不多讲了。

效果基本上已经实现了，剩下的就是修改菜单的超链接样式，在 `css.css` 中添加以下代码：

```
#menu ul li a:link,#menu ul li a:visited {font-weight:bold;color:#666}  
#menu ul li a:hover{}
```

这个也不多说了，没什么好说的了，最后的效果如下：

[首页](#) | [博客](#) | [设计](#) | [相册](#) | [论坛](#) | [关于](#)

这一节里面，主要就是想告诉大家如何使用好 **border** 和 **clear** 这两个属性。

首先，如果你曾用过 table 制作网页，你就应该知道，如果要在表格中绘制一条虚线该如何做，那需要制作一个很小的图片来填充，其实我们还有更简单的办法，只要在 `<td></td>` 中加入这么一段就可以了，你可以试试：

```
<div style="border-bottom:1px dashed #ccc"></div>
```

大家可以再次参考手册，然后你就能明白 `dashed`、`solid`、`dotted...` 等的作用，利用它们你可以制作出许多效果来，实线、虚线、双线、阴影线等等。

```
<div id="banner"></div>
```

以上代码便可以实现设计草图中的 banner，在 `css.css` 中加入以下样式：

```
#banner {  
background:url(banner.jpg) 0 30px no-repeat; /*加入背景图片*/  
width:730px; /*设定层的宽度*/  
margin:auto; /*层居中*/  
height:240px; /*设定高度*/  
border-bottom:5px solid #EFEFEF; /*画一条浅灰色实线*/  
clear:both /*清除浮动*/  
}
```

通过 **border** 很容易就绘制出一条实线了，并且减少了图片下载所占用的网络资

源，使得页面载入速度变得更快。

另一个要说明的就是 `clear:both`，表示清除左、右所有的浮动，在接下来的布局中我们还会用这个属性：`clear:left/right`。在这里添加 `clear:both` 是由于之前的 `ul`、`li` 元素设置了浮动，如果不清除则会影响 banner 层位置的设定。

```
<div id="pagebody"><!--页面主体-->
<div id="sidebar"><!--侧边栏-->
</div>
<div id="mainbody"><!--主体内容-->
</div>
</div>
```

以上是页面主体部分，我们在 `css.css` 中添加以下样式：

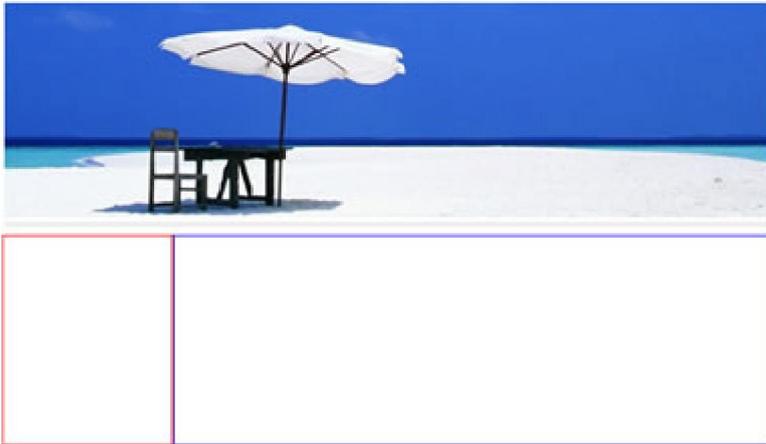
```
#pagebody {
width:730px; /*设定宽度*/
margin:8px auto; /*居中*/
}
#sidebar {
width:160px; /*设定宽度*/
text-align:left; /*文字左对齐*/
float:left; /*浮动居左*/
clear:left; /*不允许左侧存在浮动*/
overflow:hidden; /*超出宽度部分隐藏*/
}
#mainbody {
width:570px;
text-align:left;
float:right; /*浮动居右*/
clear:right; /*不允许右侧存在浮动*/
overflow:hidden
}
```

为了可以查看到效果，建议在 `#sidebar` 和 `#mainbody` 中加入以下代码，预览完成后可以删除这段代码：

```
border:1px solid #E00;
height:200px
```

保存预览效果，可以发现这两个层完美的浮动，在达到了我们布局的要求，而两个层的实际宽度应该 $160+2(\text{border})+570+2=734\text{px}$ ，已经超出了父层的宽度，由于 `clear` 的原因，这两个层才不会出现错位的情况，这样可以使我们布局的

页面不会因为内容太长（例如图片）而导致错位。



而之后添加的 `overflow:hidden` 则可以**使内容太长（例如图片）的部份自动被隐藏**。通常我们会看到一些网页在载入时，由于图片太大，导致布局被撑开，直到页面下载完成才恢复正常，通过添加 `overflow:hidden` 就可以解决这个问题。

CSS 中每一个属性运用得当，就可以解决许多问题，或许它们与你在布局的页并没有太大的关系，但是你必须知道这些属性的作用，在遇到难题的时候，可以尝试使用这些属性去解决问题。

如果您有帮助，请顶一个！文章来源：蓝色理想，感谢作者的无私奉献，现在的互联网上净化越来越少。