

[深圳网站建设]采用DIV+CSS布局的好处

1、简介：为何使用表格排版是不明智的？

表格之所以存在于 HTML 中，只是为了一个目的：显示表格状的数据。然而此后的 border="0" 使得设计师可以将图片和文本放在这无形的网格中。迄今为止，表格仍然主导着视觉丰富的的设计方式，但它却阻碍了一种更好的、更有亲和力的、更灵活的，而且功能更强大的设计方法。

2、概览：这对我有什么用处？

我们将告诉你这样一种工作方式，它会：

使你的页面载入得更快

降低你的流量费用

让你在修改设计时更有效率而代价更低

帮助你的整个站点保持视觉的一致性

让你的站点可以更好地被找到

使你的站点对浏览者和浏览器更具亲和力

在世界上越来越多人采用 Web 标准时，它还能提高你的职场竞争实力(事实上也就是降低失业的风险)。

3、表格带来的问题

把格式数据混入你的内容中。

这使得文件的大小无谓地变大，而用户访问每个页面时都必须下载一次这样的格式信息。

带宽并非免费。

这使得重新设计现有的站点和内容极为消耗劳力(且昂贵)。

这还使我们保持整个站点的视觉的一致性极难，花费也极高。

基于表格的页面还大大降低了它对残疾人和用手机或 PDA 浏览者的亲和力。

4、过渡性的设计

使用 margin 和 padding 来代替多余的表格单元和间隔 GIF。

使用 link 和 @import 来载入样式。前者用于早期浏览器，后者给现在的浏览器。

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

```
<style type="text/css" media="screen"><!--
```

```
@import url(modern.css) screen;
```

```
--></style>
```

5、结构化标记：所写即所想，所想即所写

即便书写 CSS 很简单，使用 CSS 来排版却真的需要一种和我们以往有些不同的思维方式。

我们排版时考虑的不是“这个东西放在这儿，那个东西放在那儿”，而是页面中信息的类别和信息的结构。

我们用 <h1> 标记来标示最重要的头条；次一级的条目则用 <h2> 来标记，以此类推；而段落则放在 <p> 标记中。

这就是我们称之为“结构标记”或“语义标记”的东西。

你的内容将不放在表格和表格元素中，取代它们的是 div 元素。还要给你的 div 元素安排一个 id 或 class，不过这是为了描述它们的内容或功能，而非它们的外观。

当你把某个对象设为斜体时，大概要么就是想强调它，要么是想引用某个书中标题吧？如果是前者，应该用 ; 如果是后者，其实得用 <cite>。

如果某个对象被标记为粗体，事实上它应该被标记的是，。

如果你希望在某处添加一个换行，这大概是开始了一个新的开头。如果不是开头，会不会是某种在你的站点中出现过的class (类型)？在上面两种情况下，你都应该用 CSS 替换
。

```
.foo {display:block}
```

考虑你需要导航的内容是一些无序的链接这种情况：

用 标签来编辑它们。

```
link1  
link2  
link3  
link4  
link5  
水平导航条
```

我们可以用 CSS 来控制这个列表显示在网页上的外观。

通过使用 display:inline 我们可以创建水平导航条。

```
link1  
link2  
link3  
link4  
link5
```

这是上面这个导航条用到的代码：

```
#nav1{  
margin-top: 1em;  
margin-bottom: 0.5em;  
}
```

```
#nav1 ul {  
background-color: silver;  
text-align: center;  
margin-left: 0;  
padding-left: 0;  
border-bottom: 1px solid gray  
}
```

```
#nav1 li {  
list-style-type: none;  
padding: 0.25em 1em;  
border-left: 1px solid white;  
display: inline  
}
```

```
#nav1 li:first-child {  
border: none;  
}
```

垂直导航条

```
link1  
link2
```

link3
link4
link5

这是垂直导航条的代码：

```
#nav2 {  
background-color: silver;  
border: solid 1px gray;  
width: 8em  
}
```

```
#nav2 ul {  
list-style-type: none;  
margin: 0;  
padding: 0;  
border: none  
}
```

```
#nav2 li {  
margin: 0;  
padding: 0.25em 0.5em 0.25em 1em;  
border-top: 1px solid gray;  
width: 100%;  
display: block  
}
```

```
html>body #nav2 li {  
width: auto;  
}
```

```
#nav2 li:first-child {  
border: none  
}
```

6、从玩弄表格技巧到遵循 Web 标准：通盘考虑

如果你已经把站点的内容分好了类，那么是时候分析每个页面，把它们根据逻辑关系进行分解了。

主导航条
子导航条
页眉与页脚
内容
相关信息
其他

注意分析由嵌套表格、空白分隔符和边框元素组成的表格结构 (因为我们希望把它们替换为用 div 标记组织的那种简单得多的表格结构)。

一旦你分析好页面的结构，就可以着手揭开表象，分析你现在的页面代码中可以转换成结构化标记的那些地方。

作为一个完美主义者，消灭所有的 标记和间隔 GIF 吧！

同样的，一并搞定 和
 标记。

去除表格中的那些外观标记 (bgcolor, background, 等等)。

把纯粹用来表示外观的那些 CSS 调用 (比如) 改成结构化的标记。

7、把描述外观的标记改为描述结构的标记

你可以使用“查找加替换”(或者正则表达式)，不过最好的方法还是在浏览器里打开这个页面，然后把文本复制粘贴到你的 HTML 编辑器里。

关键还是要用结构化的方式来思考！仅仅把 `` 标记替换为 `` 是远远不够的。

最重要的条目是哪个？用 `<h1>` 来标记它。次要一点的就用 `<h2>`，依此类推。用 `<p>` 来标记段落。把导航条标记为无序列表。

选定一个 DOCTYPE 来使用。(我们推荐 XHTML transitional，除非你是此道高手，否则别用 XHTML strict。)

8、把你的页面合理分布在的 div 中

把你的主导航条放在一个 id 属性设为 mainnav 的 div 中，子导航条则放在一个 id 或者 class 是 subnav 的 div 中。而页脚类似这个形式：`<div id="footer">`，至于整个页面，它被放在 `<div id="content">` 里边。

即便现在它还不怎么讨人喜欢，但只要你开始给样式表添加规则，它就会马上变得可爱起来。

9、是开始自己编写 CSS 的时候了

首先，给每个 div 设置一个边界。例如：`div {border: 1px dotted gray; padding: .5em}` 这可以帮助你完完整整地看到里边的内容，也不再需要滚动屏幕了。

首先为以定义的元素编写 CSS (`<html>`, `<body>`, `<p>`, `<h1>`, `<h2>`, ``, ``, 等等。)

尽量运用上下文相关或者有递归下降结构的选择符。这可以使得你的标记更清晰。例如：`#subnav li {border: 1px solid black; padding: .5em; display: inline}` 将能够只能影响你的子导航栏的 div 中的列表项。

在尽可能多的浏览器下进行测试，毕竟你可以让朋友帮助你用它们自己的浏览器测试

悠久是专业的深圳网站建设公司、深圳平面设计公司、深圳微信营销公司、深圳VI设计公司、深圳包装设计公司。提供网站建设、微信小程序/APP/软件开发、画册设计、包装设计、VI设计、标志设计等服务。
本文地址: <https://www.youjoys.cn/cn/news-n171.html>