

# 14.4.6 设计浮动页面

浮动布局受 HTML 原始结构的影响很大。例如，在上节示例中，如果要把次要信息列放置到页面左侧显示是非常困难的。传统方法是为主要信息列和次要信息列嵌套一个包含框，然后通过浮动实现次要信息列向左浮动，而主要信息列向右浮动的布局效果。

实现这种想法需要修改结构，修改后的结构代码如下所示：

```
<div id="main">
  <div id="submain">
    <div id="content">
      <h3>主信息区域</h3>
    </div>
    <div id="subplot">
      <h3>次信息区域</h3>
    </div>
  </div>
  <div id="serve">
    <dl>
      <dt>功能服务区域</dt>
      <dd>服务列表项</dd>
    </dl>
  </div>
</div>
```

然后，重新设计三列的浮动布局，最后所得效果如图 E14.10 所示。

```
#submain { /* 新增的内容包含框样式 */
  float:left;          /* 向左浮动 */
  width:75%            /* 内容包含框的宽度 */
#content { /* 主要信息列样式 */
  float:right;         /* 向右浮动 */
  width:55%;           /* 主要信息列宽度 */
#subplot { /* 次要信息列样式 */
  width:45%;           /* 次要信息列宽度 */
  float:left;          /* 向左浮动 */
#serve { /* 服务功能列样式 */
  width:25%;           /* 服务功能列宽度 */
  float:right;         /* 向右浮动 */
```

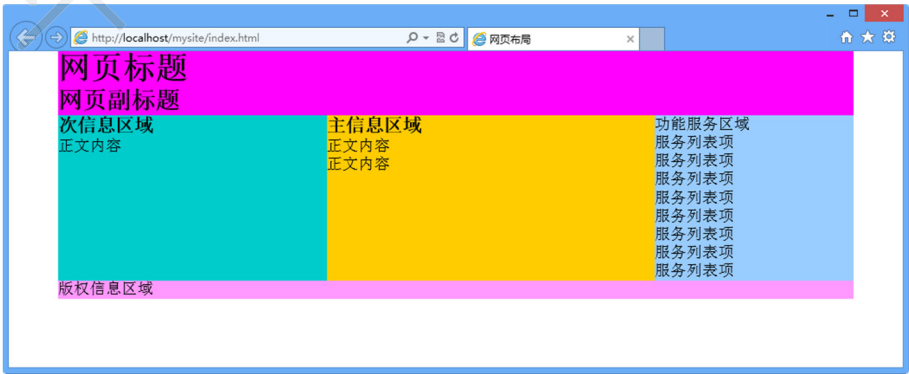


图 E14.10 通过改变结构来调整浮动列的显示位置

下面使用负 margin 的方法来实现。

**【操作步骤】**

第 1 步，为主要信息列定义 20%宽度的边界空白，这个空白专门为次要信息列备用的。

第2步，让次要信息列左边界取负值，强制其向左移动75%的距离，这个距离正好是刚定义的主要信息列的宽度和左边界之和。核心代码如下所示：

```
#content {/* 主要信息列样式 */
float:left;                /* 向左浮动 */
width:55%;                 /* 宽度 */
margin-left:20%;           /* 定义左边界，为左列留白 */
background:#FFCC00;        /* 背景色 */
}
#subplot {/* 次要信息列样式 */
width:20%;                 /* 宽度 */
margin-left:-75%;           /* 强制向左移动到主信息列的左侧 */
float:left;                /* 向左浮动 */
background:#00CCCC;        /* 背景色 */
}
```

第3步，在浏览器中预览，则所得效果如图E14.11所示。

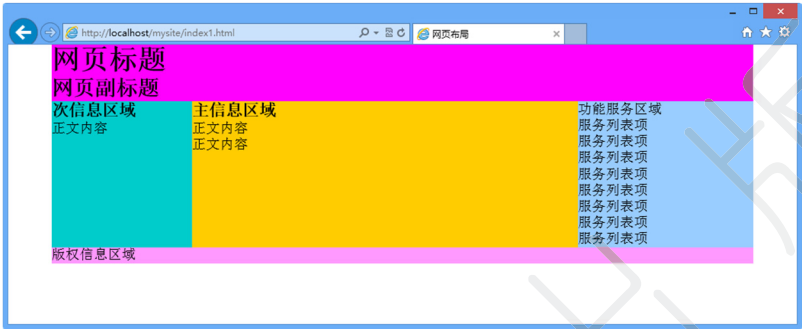


图 E14.11 通过改变结构来调整浮动列的显示位置

负边界是网页布局中比较实用的一种技巧，它能够自由移动一个栏目到某个位置，从而改变了浮动布局和流动布局存在的受限于结构的弊端，间接具备了定位布局的一些特性，当然它没有定位布局那么精确。