

JavaScript 制作菜单

1. 使用无序列表建立菜单数据

- (1) 选择“文件->打开”菜单，选择 menu.html。
- (2) 在<div id="top_right"></div>元素间输入以下内容：

```
<ul id="menu">
  <li><a href="#">产 品</a>
    <ul>
      <li><a href="#">产 品 1</a></li>
      <li><a href="#">产 品 2</a></li>
    </ul>
  </li>
  <li><a href="#">服务与下载</a>
    <ul>
      <li><a href="#">服务一</a></li>
      <li><a href="#">服务二</a></li>
      <li><a href="#">下载一</a></li>
      <li><a href="#">下载二</a></li>
    </ul>
  </li>
</ul>
```

其中，id 为 menu 的无序列表(ul)是主菜单条，menu 下的每一个 li 是主菜单项。二级 ul 即为二级菜单。

- (3) 参考第二步，请为主菜单添加主菜单项“技术支持”，并在“技术支持”下生成二级菜单：注册、投诉、联系客服和建议。

2. 通过 DHTML 控制菜单的样式

- (1) 使用 float 使主菜单项横向排列。在</head>前输入以下内容

```
<style type="text/css">
  #menu li {
    float: left;
    width: 80px;
    position: relative;
  }
</style>
```

效果如图 7-1 所示：



图 7-1 float 效果的主菜单

可以看到一级菜单项由原来的纵向排列变为了横向排列。float 属性值设为 left 将使得一级菜单项在一行中靠左停靠，因此一级菜单项将会从左到右排列。

(2) 调整主菜单条的位置和样式，在</style>之前输入下面内容：

```
#menu {
    line-height: 24px;
    list-style-type: none;
    margin: 10px 20px 20px 20px;
    font-size: 10px;
    font-weight: bold;
}
```

margin 样式为主菜单条添加外边距使得菜单移动到合适位置，效果如果 8-3 所示：



图 7-2 调整位置后的主菜单

(3) 调整主菜单项和子菜单样式，在</style>之前输入以下内容

```
#menu a {
```

```

display: block;
width: 80px;
text-align:center;
color:#FFF;
}
#menu a:link a:visited {
text-decoration:none;
}
#menu a:hover {
text-decoration:none;
font-weight:bold;
}

```

(4) 设置二级菜单条样式

```

#menu li ul {
line-height: 27px;
list-style-type: none;
text-align:left;
left: -999em;
width: 980px;
position: absolute;
float:none;
}

```

position: absolute 使得二级菜单条使用绝对定位, 该定位将二级菜单条从正常文档中移除。left: -999em 将二级菜单从屏幕中移除。由于 float 属性具有继承性, 因此需要使用 float:none 来清除主菜单项 float:left 的影响。效果如图 7-3 所示:



图 7-3 设置二级菜单样式后的效果

4. 设置其他样式。在</style>前输入如下内容:

```

#menu li ul li {
width: 80px;
background: #F6F6F6;
}

```

```

        float:none;
    }
    #menu li ul a{
        display: block;
        width: 80px;
        text-align:center;
    }
    #menu li ul a:link {
        color:#666;
        text-decoration:none;
    }
    #menu li ul a:visited {
        color:#666;
        text-decoration:none;
    }
    #menu li ul a:hover {
        color:#F3F3F3;
        text-decoration:none;
        font-weight:normal;
        background:#C00;
    }
}

```

3. 编写 JavaScript 控制菜单的显示和隐藏

- (1) 通过 javascript 访问主菜单条及主菜单项，在</head>前输入如下内容：

```

<script type=text/javascript>
function bindMenu() {
    //得到 id 为 nav 的 ul，即主菜单条
    var menuBar = document.getElementById("menu");
    //得到主菜单条的所有主菜单项
    var menuItems= menuBar.getElementsByTagName("li");
}
</script>

```

- (2) 为主菜单项编写 mouseover 事件处理代码。mouseover 事件在鼠标经过 html 元素时发生。当鼠标经过主菜单项时，主菜单项的 mouseover 事件触发，在 onmouseover 事件处理程序中将子菜单显示。在本实验步骤 2 中的第 4 小步骤中将子菜单的 left 设置为-999em 使得子菜单不在屏幕中显示，因此只要设置子菜单的 left 为 0em 就可以将子菜单显示，步骤如下：

- a. 定义子菜单显示时的样式，在</style>前输入以下样式：

```

#menu li.display_on ul {
    left: 0;
}

```

该样式指明了具有 display_on 类名的主菜单项下的子菜单 left 属性为 0，以达到显示子菜单的目的。

b. 为主菜单项编写 `mouseover` 事件处理程序，修改 `bindMenu` 函数，代码如下（粗体下划线内容为新增加）

```
function bindMenu() {  
    //得到 id 为 nav 的 ul，即主菜单条  
    var menuBar = document.getElementById("nav");  
    //得到主菜单条的所有主菜单项  
    var menuItems= menuBar.getElementsByTagName("li");  
  
    for (var i=0; i<menuItems.length; i++) {  
        //处理 mouseover 事件  
        menuItems[i].onmouseover=function() {  
            if(this.className.lastIndexOf("display_on") == -1) {  
                //通过为主菜单项添加类 display_on 达到显示子菜单的目的  
                this.className+=" display_on ";  
            }  
        }  
    }  
}
```

(3) 为主菜单项编写 `mouseout` 事件处理程序。该事件在鼠标从 HTML 元素上移走时发生，当该事件发生时需要将二级菜单再次隐藏。在第 2 步中，通过将一级菜单项设置类名 "display_on" 达到了显示二级菜单的目的，在本事件处理函数中只需要移除一级菜单项的 "display_on" 类名，即可以达到修改 `bindMenu` 函数，代码如下（粗体下划线内容为新增加）：

```
function bindMenu() {  
    //得到 id 为 nav 的 ul，即主菜单条  
    var menuBar = document.getElementById("nav");  
    //得到主菜单条的所有主菜单项  
    var menuItems= menuBar.getElementsByTagName("li");  
  
    for (var i=0; i<menuItems.length; i++) {  
        //处理 mouseover 事件  
        menuItems[i].onmouseover=function() {  
            if(this.className.lastIndexOf("display_on") == -1) {  
                //通过为主菜单项添加类 display_on 达到显示子菜单的目的  
                this.className+=" display_on ";  
            }  
        }  
        //处理 mouseout 事件  
        menuItems[i].onmouseout=function() {  
            if(this.className.lastIndexOf("display_on") == -1) {  
                //通过为主菜单项去除类 display_on 达到隐藏子菜单的目的  
                this.className = this.className.replae(" display_on ","");  
            }  
        }  
    }  
}
```

```
}  
}
```

- (4) 调用 bindMenu 函数。在</script>前编写代码：
window.onload=bindMenu;

(5) 实验思考

本实验第三大步骤可以采用 css 伪类达到相似效果，其思路是当鼠标停留在主菜单项上时(#menu li:hover)，设置二级菜单条的 left 属性。