

使用 HTML 与 CSS

本章要点

介绍 HTML 相关知识

介绍 CSS 相关知识

分析宠物分类展现的页面特点，确定开发思路

介绍宠物分类展现的静态网页版本的实现，掌握 HTML+CSS 实现静态页面的方法

通过实现宠物分类展现的静态网页版本掌握 HTML 常用标记，特别是层 `div`，超链接 `a`，图像 `img`，图像导航 `map` 和 `area`，表格 `table`、`tr`、`th`、`td` 等相关标记的用法和如何定义表单，掌握 CSS 样式表的基本语法和常用属性的用法

PetStore 的宠物分类展现部分通过实现宠物信息的展示，为用户提供宠物信息浏览功能。本章将介绍这部分功能的第一个版本——静态网页版本的实现。通过这部分的实现，让读者掌握 HTML+CSS 的相关技术。

2.1 HTML

2.1.1 HTML 文档结构

HTML 是静态网页技术，通过使用很多标签来描述网页。浏览器按照 HTML 标签的语法规则把 HTML 代码翻译成漂亮的网页。

以下为 `simple.html` 的代码：

```
<HTML>
<HEAD>
```

```
<TITLE>最简单的 HTML 文档 </TITLE>
</HEAD>
<BODY>
    Hello World
</BODY>
</HTML>
```

simple.html 是一个简单的 HTML 文档，图 2.1 为其页面效果。该文档包含了最基本的 HTML 文档结构：

- 在<HTML>与</HTML>之间放置了 HTML 文档的所有内容。
- <HEAD>至</HEAD>为 HTML 文档的开头部分，开头部分用以承载重要信息。
- <TITLE>和</TITLE>只可出现于开头部分，<TITLE>和</TITLE>之间的文本表示的是在标题栏中显示的内容，在标题栏中会显示“最简单的 HTML 文档”。
- <BODY>至</BODY>为 HTML 文档的文本部分，文本部分是实际要显示的内容，如图 2.1 所示，simple.html 在浏览器中会显示“Hello World”



图 2.1 最简单的 HTML 文档

2.1.2 HTML 标记的公共属性

HTML 标记通过设置属性来进一步控制显示的效果。表 2.1 列出了所有标记都有的公共属性及其所起的作用。

表 2.1 HTML 标记的公共属性

属 性	作 用
class	指明 HTML 标记所属的类
id	在文档中用 id 唯一地标识该标记
name	为标记命名，该名称可以在 JavaScript 或其他脚本程序中引用
style	指明标记使用的样式

2.1.3 HTML 常用标记

HTML 常用标记如表 2.2 所示。

表 2.2 HTML 常用标记

标 记	功 能	说 明
<html> </html>	创建一个 HTML 文档	
<head> </head>	设置文档标题和其他在网页中不显示的信息	
<title> </title>	设置文档的标题	
	最大的标题	还有 , , , ,
<pre> </pre>	预先格式化文本	
<u> </u>	下画线	
 	黑体字	
<i> </i>	斜体字	
 	设置字体大小从 1 到 7, 颜色使用名字或 RGB 的十六进制数值	
	创建一个段落	
<p align="">	将段落按左、中、右对齐	
 	插入一个回车换行符	
	创建一个标有数字的列表	
	创建一个标有圆点的列表	
	放在每个列表项之前, 若在 之间则每个列表项加上一个数字, 若在 之间则每个列表项加上一个圆点	
	用来排版大块 HTML 段落, 也用于格式化表	
<hr>	水平线 (设定宽度)	◆ size="..."设置线条粗细 ◆ width="..." 设置线条占据宽度 ◆ color="..."设置线条颜色
<center> </center>	水平居中	
 	创建超文本链接	◆ target="..."决定链接源在什么地方显示 (可以是用户自定义的名字, _blank, _parent, _self, _top) ◆ rel="..."发送链接的类型 ◆ rev="..."保存链接的类型 ◆ accesskey="..."指定该元素的热键 ◆ shape="..."允许使用已定义的形状定义客户端的图形镜像 (default, rect, circle, poly) ◆ coord="..."使用像素或者长度百分比来定义形状的尺寸 ◆ tabindex="..."使用定义过的 tabindex 元素设置在各个元素之间的焦点获取顺序 (使用 tab 键使元素获得焦点)
 	创建位于文档内部的书签	
 	创建指向位于文档内部书签的链接	

续表

标 记	功 能	说 明
	在网页中插入图像或视频片断	当使用 img 标记显示静态图像时，用 src 标记属性指定图像文件的 url。当使用 img 标记显示视频片断时，用 dynsrc 标签属性指定 url
<map> </map>	包含一组定义图像中链接区域的 area 元素	map 对象由 img 标记的 usemap 属性引用
<area coords=" " shape=" " url=" ">	定义图像中链接区域的形状、坐标和关联 URL	
<table> </table>	创建一个表格	
<tr> </tr>	创建表格的一行	
<td> </td>	创建表格的某行的一个单元格	
<th> </th>	创建表格的表头，表头的字是粗体显示的	
<table cellspacing="">	设置表格格子之间空间的大小	
<table border="">	设置边框的宽度	
<table cellpadding="">	设置表格格子边框与其内部内容之间空间的大小	
<table width="">	设置表格的宽度。用绝对像素值或总宽度的百分比	
<table align="">	设置表格格子的水平对齐方式（left,center,right,justify）	
<tr align="">	设置表格格子的水平对齐方式（left,center,right,justify）	
<tr valign="">	设置表格格子的垂直对齐方式（baseline,bottom,middle,top）	
<td colspan="">	设置一个表格格子跨占的列数（默认值为 1）	
<td rowspan="">	设置一个表格格子跨占的行数（默认值为 1）	

2.1.4 HTML 表单

Web 应用程序通常使用表单来获取用户的输入，所以非常重要。创建表单分为以下 2 步。

(1) 使用标记<FORM>和</FORM>来定义表单，其基本语法结构如下：

```
<FORM ACTION=url
      METHOD=get|post
      NAME=value
      ONRESET=function
      ONSUBMIT=function
      TARGET=window>

</FORM>
```

ACTION、METHOD、NAME、ONRESET、ONSUBMIT 和 TARGET 定义了表单的属性，在定义表单时，这些属性都是可选的。其中：

- ACTION 属性非常重要，其指明了表单提交时执行的动作，通常是一个服务器端

脚本程序的 URL, 比如 IncludedTop.html 里的 ../shop/searchProducts.shtml。这个 URL 一定不要错过, 否则将无法转向处理程序, 出现“无法显示网页”错误。

- **METHOD** 属性也非常重要, 表示发送表单时的 HTTP 方法 (见 1.2.1 节), 可能的值是 **post** 和 **get**, 默认是 **get**。**get** 的方式是将表单域的 **name/value** 信息经过编码之后, 通过 URL 发送 (用户可以在地址栏里看到), 而 **post** 则将表单的内容通过 **http** 发送, 用户在地址栏看不到表单的提交信息, 并且使用 **get** 方式发送信息时有 255 字符的限制。那什么时候用 **get**, 什么时候用 **post** 呢? 一般是这样来判断的: 如果只是为取得和显示数据, 而且数据量不大, 用 **get**; 一旦涉及数据的保存和更新, 那么建议用 **post**。
- **NAME** 属性表示表单名称。
- **ONSUBMIT** 和 **ONRESET** 指出单击“提交”按钮和“重置”按钮时执行的脚本代码。
- **TARGET** 用来指定显示表单结果的目标窗口或框架。

(2) 在表单中创建表单域 (或字段)。

要定义一个有用的表单还必须在 `<FORM>` 和 `</FORM>` 之间通过其他标记定义文本框、单选框、复选框、按钮和列表框等获取用户输入的表单域, 通过这些表单域, 用户可以输入文字信息或者从选项中选择, 以及进行提交的操作, 实现客户端与服务器端的交互。

HTML 表单 (Form) 的常用表单域有:

- **input type="text"**: 单行文本输入框。
- **input type="submit"**: 提交按钮, 将表单 (Form) 里的信息提交给表单里 **action** 所指向的文件。
- **input type="checkbox"**: 复选框。
- **input type="radio"**: 单选框。
- **select**: 下拉框。
- **textArea**: 多行文本输入框。
- **input type="password"**: 密码输入框 (输入的文字用 * 表示)。

2.2 CSS

CSS 的全称是 Cascading Style Sheets, 中文翻译过来就叫做层叠样式表, 又叫级联样式表, 简称样式表。使用 CSS 使 HTML 更好维护, 能简化 HTML 文档, 还能提供更强大的格式化手段。

2.2.1 CSS 分类

CSS 按其位置可以分成 3 种:

- 内嵌样式，写在标记里面通过 style 属性设置，如

```
<P style="font-size:20pt; color:red"></p>
```

这个 style 定义<p>里面的文字是 20pt 字体，字体颜色是红色。

- 内部样式表，写在 HTML 的<head></head>里面，需要使用 style 标记，如：

```
<HTML>
  <HEAD>
    <STYLE type="text/css">
      H1.mylayout {border-width:1; border:solid; text-align:center; color:red}
    </STYLE>
  </HEAD>
  <BODY>
    <H1 class="mylayout"> 这个标题使用了 Style。</H1>
    <H1>这个标题没有使用 Style。</H1>
  </BODY>
</HTML>
```

- 外部样式表。如果很多网页需要用到同样的样式，将样式写在一个以.css 为后缀的 CSS 文件里，然后在每个需要用到这些样式的网页里使用<link>标签引用这个 CSS 文件（如 Jpetstore.css）即可。本教材宠物商城项目就采用这种方式。

浏览器默认的样式优先级依次是内嵌，内部，外部。假设内嵌样式中有 font-size:30pt, 而内部样式中有 font-size:12pt, 那么内嵌式样式 font-size:30pt 就会覆盖内部样式 font-size:12pt。

2.2.2 CSS 的语法

一个样式的语法由三部分构成：选择器（Selector）、属性（Property）和属性值（Value），如图 2.2 所示。其中，P 就是选择器，color 就是属性，red 就是属性值。

如果想为 Style 加多个属性，在两个属性之间要用分号加以分隔。下面的 Style 就包含 2 个属性，一个是对齐方式居中，一个字体颜色为红，当中用分号分隔开。

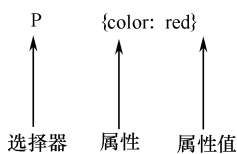


图 2.2 样式的语法

```
p {text-align:center;color:red}
```

为了提高 Style 代码的可读性，也可以分行写：

```
p{
    text-align: center;
    color: black;
    font-family: arial
}
```

本教材采用的案例系统的样式文件 jpetstore.css 就是采用分行写的方式。

2.2.3 CSS 的选择器

CSS 中最常用的有 5 类选择器。

- 标记选择器：HTML 中所有的标记。
- id 选择器：即使用标记的 id 属性值前面加上#作为选择器，如宠物商城中的层 `<div id="Footer">`，然后在样式表中定义样式。

```
#Footer {
    width: 99%;
    float:left;
    background-color: #000
}
```

- 类别（class）选择器：在 CSS 里用一个点开头表示类别选择器定义。类别选择器可以为相同的 HTML 标记设置不同的样式。比如说，希望段落

有两种样式：一种是居中对齐，一种是居右对齐。可以写如下样式：

```
p.center {text-align:center}
p.right {text-align:right}
```

其中 center 和 right 就是两个 class。可以引用这两个 class，示例代码如下：

```
<p class="center">这一段居中显示。</p>
<p class="right">这一段是居右显示。</p>
```

- 群组选择器：当几个元素样式属性一样时，可以共同调用一个声明，元素之间用逗号分隔。如：

```
h1,h2,h3,h4,h5,h6 {
    color: red
}
```

上面的例子是将所有正文标题(<h1>到<h6>)的字体颜色都变成红色。

- 后代选择器：后代选择器也叫派生选择器。可以使用后代选择器给一个元素里的子元素定义样式，例如宠物商场样式文件 jpetstore.css 中：

```
#Menu a {
    color: #eaac00;
    text-decoration: none
}
```

- #Menu a 表示为 Menu 层中的超链接（a）定义颜色和下画线样式。

2.2.4 CSS 的伪类

CSS 中用四个伪类：a:link、a:visited、a:hover 和 a:active 来定义链接的样式，分别定义“链接、已访问过的链接、鼠标停在上方时、单击鼠标时”的样式。如：

```
a:link {font-weight : bold ;text-decoration : none ;color : #c00 ;}  
a:visited {font-weight : bold ;text-decoration : none ;color : #c30 ;}  
a:hover {font-weight : bold ;text-decoration : underline ;color : #f60 ;}  
a:active {font-weight : bold ;text-decoration : none ;color : #f90 ;}
```

以上语句分别定义了“链接、已访问过的链接、鼠标停在上方时、单击鼠标时”的样式。注意，必须按以上顺序写，否则显示可能和你预期的不一致。记住它们的顺序是“LVHA” (Link, Visited, Hover, Active)。

2.2.5 CSS 的盒子模式

为了能更好地理解 CSS 的属性，需要理解 CSS 中盒子模式（box model）的概念。如图 2.3 所示，黑框包围的一个方块，就是一个盒子（box），对应定义样式的区域。

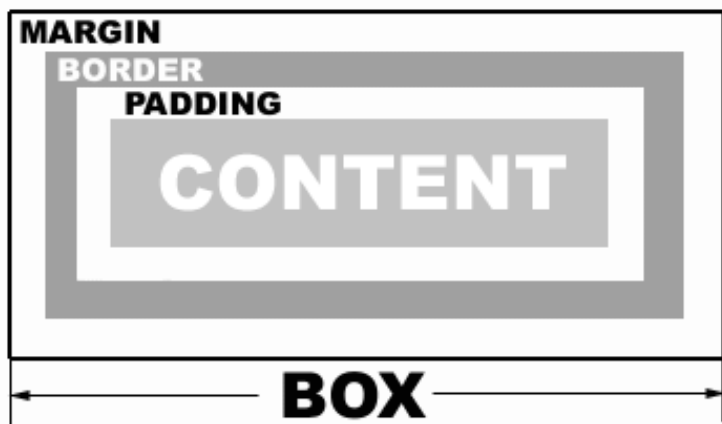


图 2.3 盒子模式示意图

盒子里由外至里依次是：

- margin 边距。
- border 边框。
- padding 间隙（也有人称做补丁）。
- content（内容，比如文本，图片等）。

CSS 边距属性（margin）用来设置一个元素所占空间的边缘到相邻元素之间的距离。

CSS 边框属性（border）用来设定一个元素的边线。

CSS 间隙属性（padding）用来设置元素内容到元素边框的距离。

CSS 背景属性（background）指的是 content 和 padding 区域。

CSS 属性中的 width 和 height 指的是 content 区域的宽和高。

2.2.6 CSS 的常用属性

表 2.3 为 CSS 常用属性列表。

表 2.3 CSS 常用属性

属 性	名 称	例
左边距属性	margin-left	.d1 {margin-left:1cm}
右边距属性	margin-right	.d1 {margin-right:1cm}
上边距属性	margin-top	.d1 {margin-top:1cm}
下边距属性	margin-bottom	.d1 {margin-bottom:1cm}
边距属性(以上 4 个属性的综合)	margin	.d1 {margin:1cm} .d1 {margin:1cm 2cm 3cm 4cm}
边框宽度属性	border-width	.d1 {border-width:10px;
边框风格属性	border-style	.d1 {border-style:solid;}
边框颜色属性	border-color	.d5 {border-color:gray; }
边框属性(以上 3 个属性的综合)	border	.d5 {border-color:gray;border-style:solid;}
左间隙属性	padding-left	.d1 {padding-left:1cm}
右间隙属性	padding-right	.d1 {padding-right:1cm}
上间隙属性	padding-top	.d1 {padding-top:1cm}
下间隙属性	margin-bottom	.d1 {padding-bottom:1cm}
间隙属性(以上 4 个属性的综合)	padding	.d1 {padding:1cm} .d1 {padding:1cm 2cm 3cm 4cm}
颜色属性	color	.p1 {color:gray}
行间距	line-height	.p2 { line-height: 20px;}
背景颜色属性	background-color	body {background-color:#99FF00;}
字体名称属性	font-family	.s1 {font-family:Arial}
字体大小属性	font-size	.s2 {font-size:16pt}
字体风格属性	font-style	.s1 {font-sytle:italic}
字体浓淡属性	font-weight	<p style = "font-weight:bold">这段文字字体的浓淡属性(font-weight)值是 bold。<p>
字体属性	font	.s1 {font:italic normal bold 11pt arial}
文本对齐属性	text-align	.p2 {text-align:right}
设定文本画线的属性	text-decoration	a.Button, a.Button:link, a.Button:visited { padding: 3px; color: #fff; background-color: #005e21; text-decoration: none; }
文本, 图片等的宽度属性	width	img {width:99%;}
文本, 图片等的高度属性	height	img { height:11px;}
漂浮属性	float	#Footer {float:left;}
表格边框样式	border-collapse	table { border-collapse:collapse; border-sapcing: 10px, 50px; caption-side: bottom; empty-cells: show; table-layout: auto; }
表格边框距离	border-spacing	
表格标题对齐	caption-side	
空单元格是否显示	empty-cells	
单元格大小自适应	table-layout	

2.3 宠物分类展现的页面及 Web 应用开发步骤

2.3.1 宠物分类展现的页面

系统的主页面展示宠物分类，如图 2.4 所示。



图 2.4 主页面

单击导航条、快捷菜单的相应商品分类以及相应图片链接，如鱼，将进入这类（Category）宠物的 Product 列表页面，在这个页面会列出所有此分类的产品的不同品种，如图 2.5 所示。可通过返回主菜单链接回到主页面。



图 2.5 同一 Category 的 Product 列表页面

单击其中的一个商品编号链接，如 FI-FW-01，将进入此 Product（这里是锦鲤）的所有 Item 列表（这里是斑点锦鲤和无斑点锦鲤 2 项）页面，如图 2.6 所示。

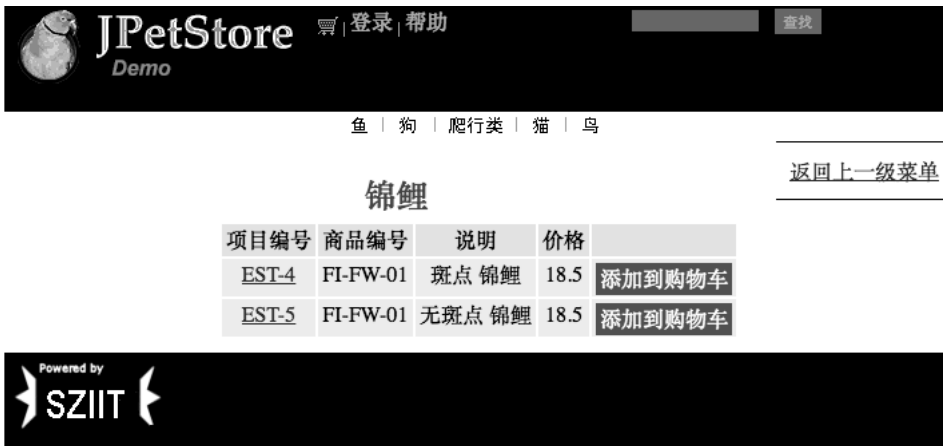


图 2.6 同一 Product 的 Item 列表页面

在 Item 列表页面单击其中的一个 Item，如 EST-4 对应的“添加到购物车”链接，可将对应 Item（这里是斑点锦鲤）加入购物车，也可单击该 Item 项目编号链接，如 EST-4 进入该 Item 的详细信息页面，如图 2.7 所示。



图 2.7 Item 的详细信息页面

2.3.2 使用 MyEclipse 开发 Web 应用的步骤

MyEclipse 是开发 Java Web 项目的利器。用 MyEclipse 开发 Web 应用的步骤如下。

1. 创建一个 Web 项目

在 MyEclipse 中，在菜单栏依次选择“File”→“new”→“Web Project”启动 Web 项目创建向导，可以很容易地创建一个 Web 项目，如图 2.8 所示。

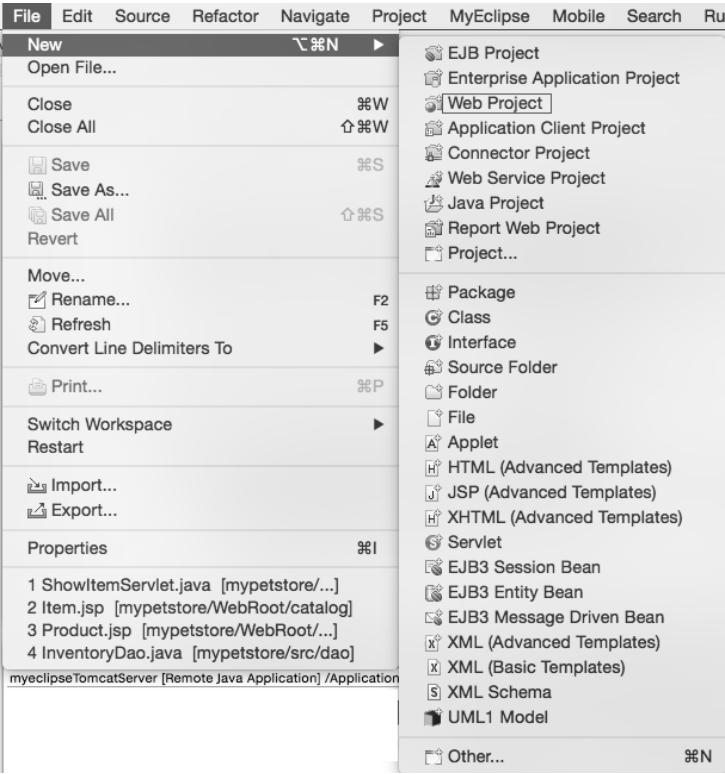


图 2.8 创建一个 Web 项目

2. 设计 Web 项目的目录结构

当在 MyEclipse 中创建完毕一个新的 Web 项目后，就可以在 MyEclipse 的“包资源管理器”中看到这个 Web 项目的目录结构，它由 MyEclipse 自动生成，如图 2.9 所示。

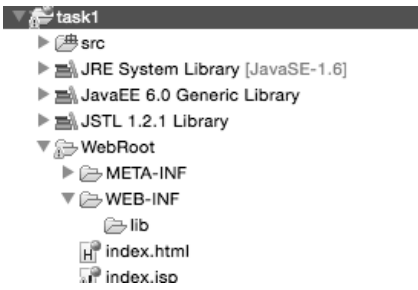


图 2.9 Web 项目的目录结构

其中，src 存放 Java 源文件，WebRoot 是这个 Web 应用项目的文档根目录，由以下部分组成。

- META-INF 目录：由系统自动生成，存放系统描述信息。
- WEB-INF 目录：该目录下存放的文件不能被用户访问到。该目录主要包括 lib 目录和 web.xml 文件。lib 目录存放 Web 应用所需要的.jar 文件或者.zip 文件，如 MySQL 数据库的驱动开发包；web.xml 文件是 Web 应用的配置文件，非常重要。

- 静态文件：包括所有 HTML 文件、CSS 文件、图像文件等，如 index.html。
- JSP 文件：如 index.jsp。

3. Web 项目编码及调式

以创建 HTML 页面为例，在 MyEclipse 中为某个 Web 项目创建 HTML 页面并调试的步骤如下：

- （1）使用 HTML 向导或文件向导在 WebRoot 或其某个子文件夹中创建一个新的 HTML 文件。
- （2）按照需求在其中编写 HTML 页面代码。
- （3）查看 HTML 页面效果。需要先启动对应 Web 项目（右击该项目，选择 Run As → MyEclipse Server Application 命令，如图 2.10 所示），然后在打开的浏览器的地址栏录入该 HTML 文件的 url，按回车键即可查看页面效果。

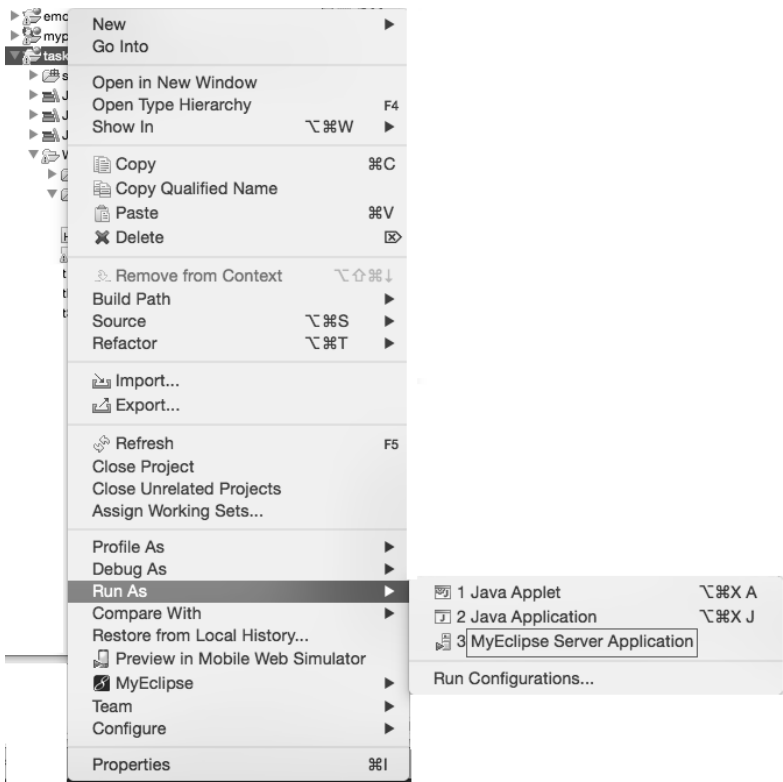


图 2.10 在 MyEclipse 中启动 Web 项目

2.4 宠物商城术语表

为了更好地描述，本教材使用如表 2.4 所示的术语。

表 2.4 宠物商城术语表

术 语	说 明
Category	宠物所属分类，本系统提供出售的宠物分属鱼、狗、猫、爬行动物和鸟类
Product	同一分类的宠物可能有不同 Product，如鱼就有锦鲤、金鱼、天使鱼、虎鲨这些品种
Item	同一品种的宠物可能有不同 Item，如锦鲤就有有斑点的锦鲤和无斑点的锦鲤 2 个系列，用户最终是要了解某 Item 的宠物的详细情况

本系统分层次展示宠物信息：

- （1）列出用户选择的某 Category 的所有 Product （见文件 Category.html 或 Category.jsp）。
- （2）再列出用户选择的某 Product 的所有 Item （见文件 Product.html 或 Pruduct.jsp）。
- （3）最后列出用户选择的某 Item 的详细信息（见文件 Item.html 或 Item.jsp）。

2.5 实现主页面 Main.html

宠物分类展现的静态版本采用 HTML 和 CSS 技术实现，体现了“先考虑显示内容正确，再考虑美化页面”的实现思路：HTML（包括层 DIV、超链接、表格、图像、换行等常用标记标记）完成显示内容，然后使用 CSS 对页面效果进行设置。如图 2.4~图 2.7 所示 4 个页面具有公共的头部和底部，变化的只是中间部分（主体），本章只考虑主体的实现，公共的头部和底部暂时不考虑。

主页面 Main.html 存放在\mypetstore\WebRoot\catalog 目录下。

2.5.1 主页面的左边导航条部分代码

主页面的左边导航条部分，即图 2.4 的左边部分，如图 2.11 所示。

分析图 2.11，可以发现，只要实现了鱼类相关的导航，其他的导航可以仿照实现。“鱼”不是文本是一个图片，对应文件是../images/fish_icon.gif，“..”表示当前文件所在目录（文件夹）的上一级目录（文件夹），即如果 Main.html 在\mypetstore\WebRoot\catalog 目录下，则 fish_icon.gif 在\mypetstore\WebRoot\images 目录下。“海水鱼，淡水鱼”是文本，前面有 2 个空格。

鱼类相关的导航部分代码如下：

```
<A href="Category.html"> </A>
<br>
    &nbsp;   海水鱼，淡水鱼
```

- 鱼
海水鱼，淡水鱼
- 狗
不同品种的狗
- 猫
不同品种、珍稀品种的猫
- 爬行类
蜥蜴，龟类，蛇类
- 鸟类
珍稀品种的鸟

图 2.11 主页面的左边导航条部分

上面代码用到以下 HTML 标记：

- 显示图片，表示在页面显示../images/fish_icon.gif 表示的图像。
- 超链接，如，是一个图像链接，单击图片将转向页面 Category.html。
- 为一个空格符。
-
 为换行符。

狗、猫、爬行类和鸟类对应图片文件分别是 dogs_icon.gif、cats_icon.gif、reptiles_icon.gif 和 birds_icon.gif。请补充完成狗、猫、爬行类和鸟类的导航代码，可得到如图 2.12 所示页面效果。

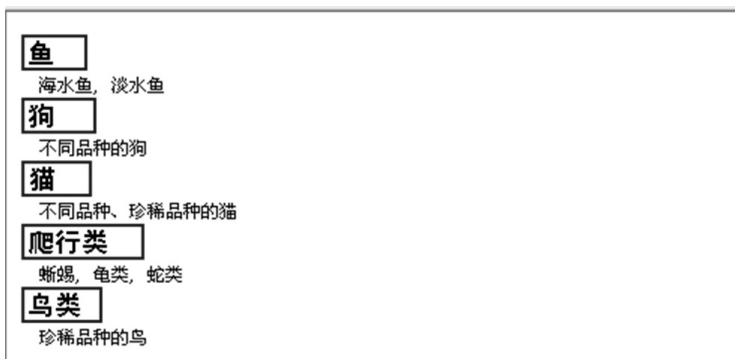


图 2.12 导航条部分代码页面效果

2.5.2 主页面的图片导航代码

以下为主页面的图片导航部分的代码，注意只实现了鱼类的图片导航。

```
<map name="estoremap">
  <area alt="Fish" coords="2,180,72,250" href="Category.html"
    shape="RECT"/>
  <!--在此补齐狗、爬行类、猫和鸟类的图片导航代码-->
</map>

```

图片导航通过在 map 元素中使用 area 元素及其 coords 值和 shape 标签属性来实现。具体包括：

- map 只作为 area 的容器，包含一组定义图像中链接区域的 area 元素。
- <area>定义可以当作导航用的区域，就是用来定义 Dreamweaver 中的热点。如<area alt="Fish" coords="2,180,72,250" href="Category.html" shape="RECT"/> shape 属性指明了区域的形状为矩形，coords 指明了形状边界的左上角和右下角的坐标

(2,180) 和(72,250), 见图 2.13。href 为导航到的页面, alt 属性可设置替换用的文字, 针对那些无法显示 area 的浏览器。area 不能独立出现, 必须嵌套在 map 中, 即<map>与</map>之间。

- 通过在标记中增加属性 usemap 的定义 (这里是#estoremap), 说明该图片提供 name 为 estoremap 区域导航。增加 height 属性设置图片的高度。增加 align 属性设置图片的对齐方式。

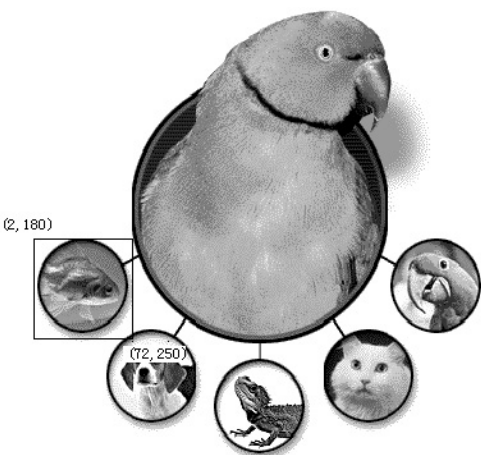


图 2.13 鱼图片导航坐标示意

如果狗、爬行类、猫和鸟类热点区域的左上角和右下角的坐标如表 2.5 所示, 请补齐狗、爬行类、猫和鸟类的图片导航代码, 可得到如图 2.14 所示页面效果。

表 2.5 狗、爬行类、猫和鸟类热点区域的左上角和右下角的坐标

宠物分类图片	左上角坐标	右下角坐标
狗	(60,250)	(130,320)
爬行类	(140,270)	(210,340)
猫	(225,240)	(295,310)
鸟	(280,180)	(350,250)

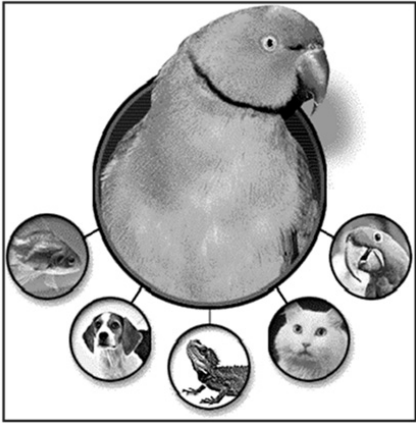


图 2.14 图片导航代码页面效果

2.5.3 通过层 DIV 标记对主页面 Main.html 进行布局

<div>和</div>定义一个层，在里面可以放任何需要展示的内容，可以嵌套使用，也就是层里面还可以有层。主页面的 Main.html 使用 div 进行布局，定义 1 个层：Main，主页面的主体部分整个放在 Main 层中，在 Main 层中定义 2 个子层：Sidebar 和 MainImage，分别存放导航条和图片导航，即：

```
<div id="Main">
  <div id="Sidebar">
    导航条代码
  </div>
  <div id="MainImage">
    图片导航代码
  </div>
</div>
```

以上代码将得到如图 2.15 所示页面效果。

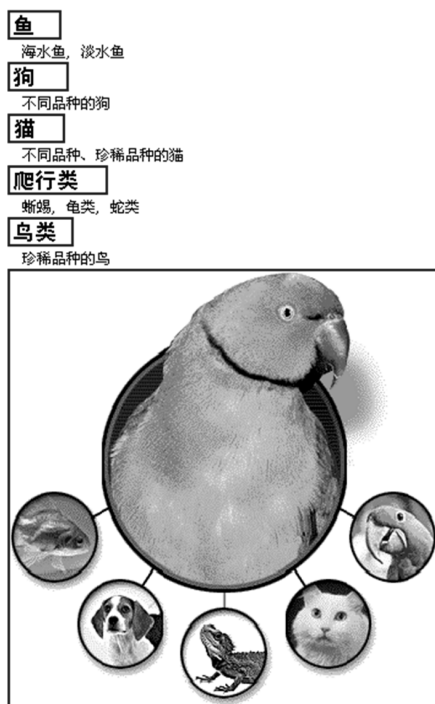


图 2.15 未设置格式的 Main.html 页面效果

2.5.4 通过 CSS 设置效果

跟目标系统比，有以下问题需要解决：

- 不能显示图片边框。
- 2 个层（即 MainImage 和 Sidebar 层）位置需要调整。

在\mypetstore\WebRoot\css 目录下建立样式表文件 jpetstore.css。

```
img {  
    border: 0;  
}  
#Main {  
    height: 100%;  
    width: 99%;  
    background-color: #FFF;  
}  
#Sidebar {  
    float: left;  
    background: inherit;  
    width: 30%;  
}  
#MainImage {  
    float: left;  
    background: inherit;  
    text-align: center;  
    width: 50%;  
}
```

设置图片无边框(border 值为 0)

设置 Sidebar 层靠父层左边停靠 (float), 继承 (inherit) 父层背景色, 占父层 30%的宽度, 注意 Sidebar 前面有# 是一个 id 选择器

设置 MainImage 层靠左停靠, 继承父层背景色, 占 50%的宽度, 文字居中对齐

在 Main.html 前添加下面语句引入样式表:

```
<Link Rel="STYLESHEET" Href="../css/jpetstore.css" Type="text/css">
```

再打开 Main.html 文件, 会看到如图 2.16 所示页面效果。

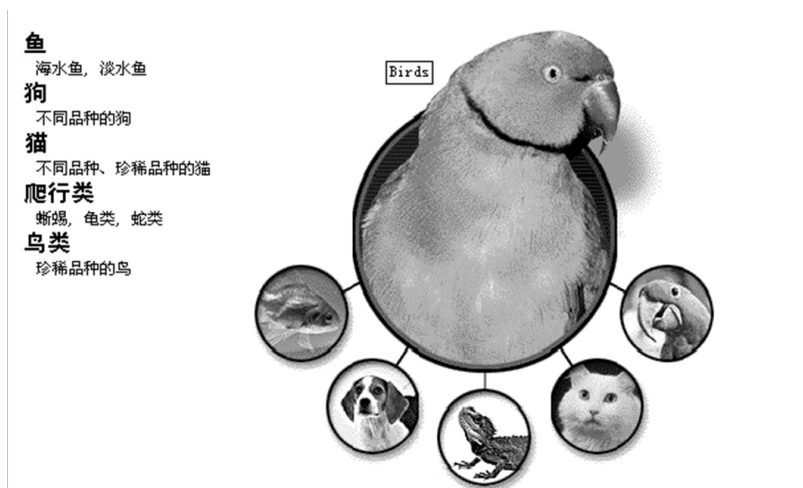


图 2.16 用样式表设置格式的 Main_body.html 页面效果