



火龙果·整理
uml.org.cn

HTML设计与应用

无兄弟
不编程



HTML设计与应用

1. 网页制作概述
2. XHTML语言的语法
3. XHTML文件的主体结构
4. 文字版面的编辑
5. 创建图像和链接
6. 使用XHTML表格
7. XHTML框架结构
8. XHTML表单设计格式标签



一、网页制作概述

- ❖ 1.1 什么是HTML?
- ❖ 1.2 HTML的历史
- ❖ 1.3 什么是XHTML?



1.1 什么是HTML

- ❖ HTML是用来制作网页的标记语言
- ❖ HTML是Hypertext Markup Language的英文缩写，即超文本标记语言
- ❖ HTML语言是一种标记语言，不需要编译，直接由浏览器执行
- ❖ HTML文件是一个文本文件，包含了一些HTML元素，标签等。HTML文件必须使用html或htm为文件名后缀
- ❖ HTML是大小写不敏感的，HTML与html是一样的
- ❖ HTML是由W3C的维护的



HTML 示例

```
<html>
  <head>
    <title>本页面的标题</title>
  </head>

  <body bgcolor="black" text="#ffffff">
    <center>
      <h1>这是第一个HTML实例</h1>
    </center>
    <hr width="80%">
    <p>本页显示黑色背景，白色的文本</p>
  </body>
</html>
```

```
<!-- HTML开始标记 -->
<!-- HTML头信息开始标记 -->
<!-- 网页标题标记 -->
<!-- HTML头信息结束标记 -->
<!-- HTML中可以使用空行调整编码格式-->
<!-- 网页主体标记 -->
<!-- HTML格式标记中的居中标记-->
<!-- HTML内容1号标题标记-->
<!-- HTML居中格式的结束标记-->
<!-- HTML中输出分隔线标记 -->
<!-- HTML段落的标记对-->
<!-- 页面体中内容结束标记 -->
```





1.2 HTML的历史

- ❖ HTML 1.0 -- 1993年6月, IETF发布
- ❖ HTML 2.0 -- 1995年11月, 发布
- ❖ HTML 3.2 -- 1996年1月, W3C推荐标准
- ❖ HTML 4.0 -- 1997年12月, W3C推荐标准
- ❖ HTML 4.01 -- 1999年12月, W3C推荐标准
- ❖ HTML 5.0 -- 2008年8月, W3C工作草案



1.3 什么是XHTML?

- ❖ XHTML 指扩展超文本标签语言 (EXtensible HyperText Markup Language)。
- ❖ XHTML 与 HTML 4.01 几乎是相同的,可以说就是HTML一个升级版。
- ❖ XHTML 是更严格更纯净的 HTML 版本。
- ❖ XHTML 是作为一种 XML 应用被重新定义的 HTML。
- ❖ XHTML 是一个 W3C 标准,基于XML的应用,语言更简洁更严谨。
- ❖ XHTML 的目标是取代 HTML,建立XHTML的目的就是实现HTML向XML的过渡。
- ❖ XHTML是大小写敏感的,XHTML与xhtml是不一样的.标准的XHTML标签应该使用小写。



二、XHTML语言的语法

- ❖ 2.1 HTML标签
- ❖ 2.2 HTML标签属性
- ❖ 2.3 HTML元素
- ❖ 2.4 HTML文档
- ❖ 2.5 HTML注释
- ❖ 2.6 特殊字符

2.1 HTML标签

- ❖ 标签是HTML中最基本单位,也是最重要组成部分。
- ❖ 通常要用两个角括号括起来:“<”和“>”。
- ❖ 标签都是闭合的(两种形式:成对与不成对)

<标签名>内容</标签名> 如: <table></table>
<标签名 /> 如:

- ❖ 标签是大小写无关的,<body>跟<BODY>表示意思是一样的,标准推荐使用小写,这样符合XHTML标准

常用HTML标签

html -- 定义html文档
head -- 定义文档头信息head
a -- html链接标签a
div -- html层div
tr -- 定义表格行tr
form -- html表单标签form

body -- 定义文档体body
title -- 定义文档的标题title
img -- html图像标签img
table -- 定义html表格table
td -- 定义表格列td
input -- 定义表单的输入域input



2.2 HTML标签属性

- ❖ HTML属性一般都出现在HTML标签中，是HTML标签的一部分。
- ❖ 标签可以有属性，它包含了额外的信息。属性的值一定要在双引号中。
- ❖ 标签可以拥有多个属性。
- ❖ 属性由属性名和值成对出现。
- ❖ 语法：

```
<标签名 属性名1="属性值" 属性名2="属性值" ... 属性名N="属性值"></标签名>
```

- ❖ 示例：

```
<a href="http://www.lampbrother.net">兄弟连网址</a>
```

```
<!-- 定义一个兄弟连网站的超级链接 -->
```

```

```

```
<!-- 插入一张“linux_logo.jpg”图片，并设置图片显示的高度和宽度 -->
```

2.3 HTML元素

- ❖ HTML元素是构建网页的一种单位,是由HTML标签和HTML属性组成的,HTML元素也是网页中的一种基本单位。
- ❖ HTML文档就是由元素和标记组成的文件。
- ❖ 示例:

```
<a href="http://www.lampbrother.net">  
    兄弟连网址  
</a>
```

这是一个HTML链
接元素

```
<p>  
    这是我的第一个网站,欢迎来到  
    <a href="http://www.lampbrother.net">  
        兄弟连  
    </a>社区。  
</p>
```

这是一个HTML
段落元素,它包含
了一个HTML链
接元素.



2.4 HTML文档

- ❖ HTML文档就是HTML页面，也就是网页，是由HTML元素组成的
- ❖ 互联网的所有内容都是由一个个HTML文档体现的，如右侧页面。





2.5 HTML注释

HTML注释, 我们经常要在一些代码旁做一些HTML注释, 这样做的好处很多, 比如: 方便查找, 方便比对, 方便项目组里的其它程序员了解你的代码, 而且可以方便以后你对自己代码的理解与修改等等.

语法: <!-- 被注释的内容 -->

```

```

```
<!-- 使用alt属性的图片 -->
```

```

```

```
<!-- 设置图片的边框宽度2px -->
```

```

```

```
<!-- 只设置宽度高度自动适应 -->
```

```

```

```
<!-- 图片高度和宽度一起设置 -->
```

2.6 特殊字符

❖ 特殊字符：用数值码和引用实体两种方式表示

■ 例如：

`©` `©`

在什么情况下需要用`"`表示？

代码	效果	代码	效果
<code>&quot;</code>	“	<code>&amp;</code>	&
<code>&lt;</code>	<	<code>&gt;</code>	>
<code>&nbsp;</code>	空格	<code>&copy;</code>	©
<code>&sect;</code>	§	<code>&curren;</code>	¤

❖ 如果属性值中同时有单引号和双引号，该如何处理呢？假设`<p>`标签的`title`属性值为 `'z' x" x`，那么，

■ 不使用引号的写法为：`<p title= 'z' x" x></p>`

■ 使用单引号的写法为：`<p title= ' 'z' x" x' ></p>`有错

■ 使用双引号的写法为：`<p title= " 'z' x" x" ></p>`有错

■ 特殊字符的写法为：`<p title= " 'z' x"x" ></p>`正确

❖ 思考题：

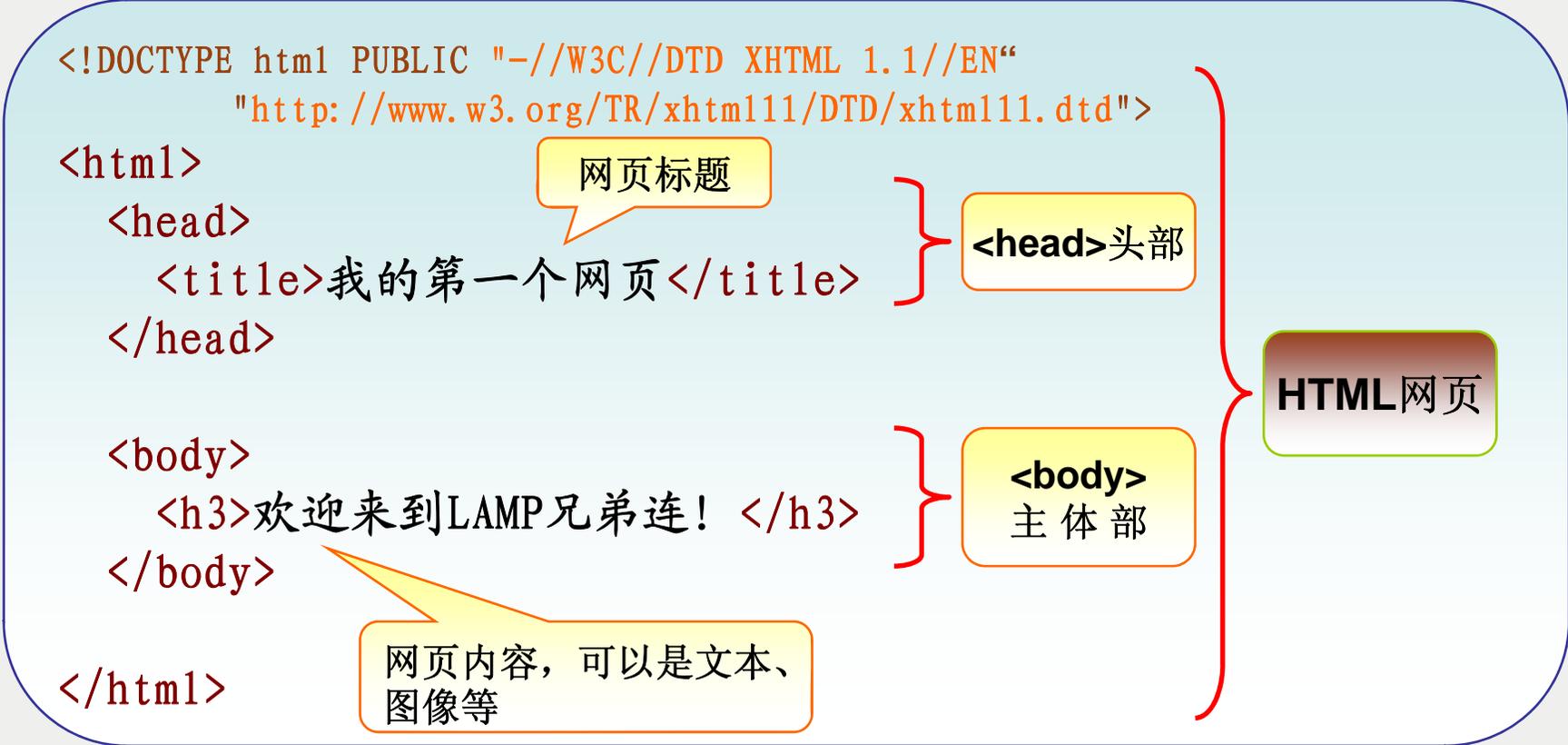
■ 人们为什么要对`&`、`<`、`>`字符安排一个HTML编码文本代码呢？



三、XHTML文件的主体结构

- ❖ 3.1 HTML文档的结构
- ❖ 3.2 HTML head 标签
- ❖ 3.3 HTML规范与版本
- ❖ 3.4 DTD文档类型

3.1 HTML文档的结构



❖ 文档结构中的标签

- <html> </html>
- <head> </head>
- <title></title>
- <body> </body>



❖ **html** 标签 -- 代表HTML文档的开始

```
<html xmlns="http://www.lampbrother.net" xml:lang="zh-CN" dir="ltr">
```

❖ 说明:

- **xmlns**: 代表使用http://www.lampbrother.net做为命名空间
- **xml:lang="zh-CN"**: 代表使用中文作为页面文字
- **dir="ltr"**: 代表左到右(left-to-right)的文字对齐方式

❖ **body** 标签 -- 代表HTML文档的主体, 常见属性如下。

```
<body text="#ff0000" bgcolor="#ffffee7" link="#3300ff" vlink="#9900ff">
```

❖ 说明:

- **text**: 设定页面文字的颜色
- **bgcolor**: 设定页面背景的颜色
- **link**: 设置页面默认的连接颜色
- **vlink**: 设定访问后的连接颜色
- **alink**: 设定鼠标正在单击时的连接颜色
- **background**: 设定页面背景图像
- 还有**class**、**name**、**id**、**style**等属性。



3.2 HTML head 标签

- ❖ head标签代表HTML文档的头信息，以<head>开始，</head>结束。
- ❖ 头(head)包含了当前文档的一些信息，例如标题信息，meta信息等，正常情况下头信息是不会显示在HTML文档中的。
- ❖ head元素包含的标签：

- ❖ **title**标签 -- 代表HTML文档的标题
- ❖ **base**标签 -- 定义基URL用于页面的链接与引用
- ❖ **link**标签 -- 当在文档中声明使用外接资源(比如CSS)时使用此标签
- ❖ **style**标签 -- 在文档中声明样式时使用此标签
- ❖ **script**标签 -- 在文档中使用JavaScript脚本
- ❖ **meta**标签 -- 可以为HTML文档提供额外信息：编码，作者，版权与关键字



```
<head>
  <base href="http://www.lampbrother.net">
  <title>LAMP兄弟连--易第优教育|PHP|PHP培训|LAMP|LAMP培训</title>
  <meta http-equiv="Content-Type" content="text/html;
    charset=gb2312" />
  <meta name="keywords" content="PHP培训|LAMP培训|就业培训|">
  <meta name="description" content="LAMP兄弟连--致力于PHP培训" />
  <meta name="author" content="lampbrother, gaoluofeng">
  <script type="text/javascript" src="js/pngforie6.js"></script>
  <link type="text/css" rel="stylesheet" href="./css/index.css"/>
  <style>
    body {
      margin: 0px;
      padding: 0px;
      background: red;
      color: green;
    }
  </style>
</head>
```



3.3 HTML规范与版本

- ❖ 不同的浏览器之间的不兼容问题
- ❖ 为什么要制定规范
- ❖ 谁来制订规范
 - IETF
 - W3C
- ❖ 正在使用的HTML版本
 - HTML4.01
 - XHTML



IETF与W3C介绍

❖ IETF

- Internet Engineering Task Force (Internet工程任务组)，由网络设计人员、操作员、厂商、专家组成的民间组织，主要负责有关Internet的各种技术标准及接口规范的制定，其网址为<http://www.ietf.org>。IETF以RFC (Requests for Comments, 请求注解文档) 定名所议，例如RFC2616 (HTTP/1.1)、RFC1521 (MIME)。

❖ W3C

- World Wide Web Consortium (习惯称之为WWW联盟) 于1994年成立，是与WEB有关的企业机构成立的业界同盟，其网址为<http://www.w3c.org>。W3C目前的成员仅限于团体或组织，只要交纳一定的费用，并签署一份保证遵守规则的成员协议，任何公司均可加入，W3C对WEB的标准握有生杀大权，负责研究、审定、发布、管理有关WEB的标准，例如，HTML、CSS等。



3.4 DTD文档类型

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

- ❖ DTD: (Document Type Definition, DTD)
- ❖ `<!DOCTYPE HTML PUBLIC "version name" "url">`
 - HTML指定文档类型的名称, 为了说明这个页面使用了何种HTML规则, 或者结构。
 - PUBLIC表明所依据的DTD文件可对任何人公开访问, 而不是某个公司内部的规范文件。
 - version name指定该HTML版本的标识名称。例如, HTML4.0的标识名称为“-//w3c//DTD HTML 4.01 Transitional//EN”。对于ISO标准的DTD以“ISO”三个字母开头; 被改进的非ISO标准的DTD以加号“+”开头; 没被改进的非ISO标准的DTD以“-”开头。
 - url指定该HTML语言的定义规范文件在internet上的位置, 例如, <http://www.w3.org/TR/html4/loose.dtd>。
 - 规范文件: loose.dtd、strict.dtd、frameset.dtd等文件名的区别。



四、文字版面的编辑

- ❖ 4.1 格式标签
- ❖ 4.2 文本标签



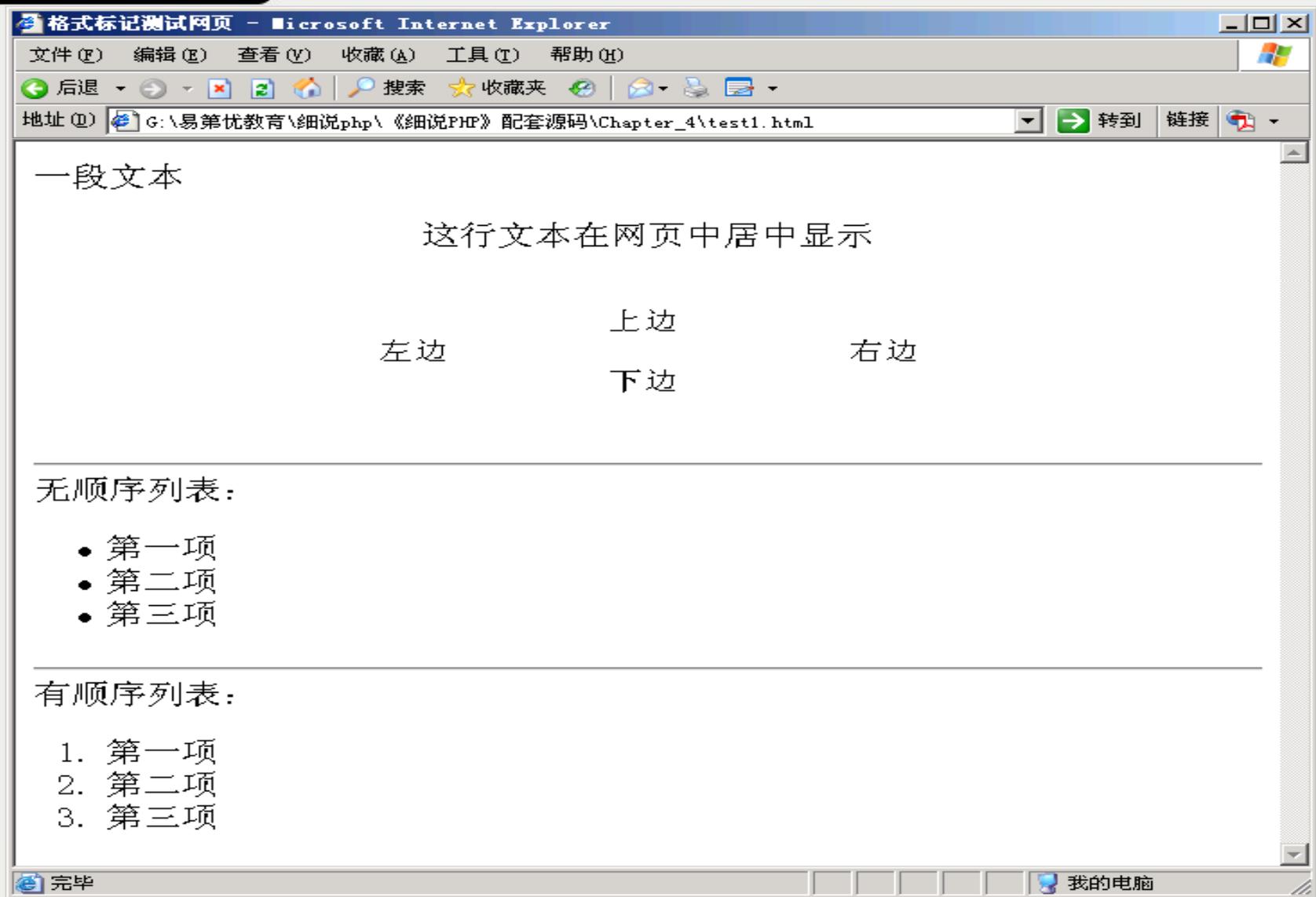
4.1 格式标签

标签	描述
 	换行标签，完成文字的紧凑显示。可以使用连续多个 标签来换行
<p>	段落标签，里面可以加入文字，列表，表格等，可以<p></p>或<p />使用
<center>	居中对齐标签，使段落或文字相对于上一层标记居中显示
<pre>	按原文显示标签，可以把原文件中的空格，回车，换行，tab键表现出来
	代表HTML列表项目，每个列表项使用一对标记
	代表HTML无序列表，里面每一列表项使用标签定义
	代表HTML有序列表，里面每一列表项使用标签定义
<hr>	水平分割线标签，用于段落与段落之间的分割



```
<html>
<head><title>格式标记测试网页</title></head>
<body>
  <p>一段文本</p>
  <center>这行文本在网页中居中显示</center>
  <pre>
    上边
  左边      右边
    下边
  </pre>
  <hr>
  无顺序列表:
  <ul>
    <li>第一项</li>
    <li>第二项</li>
    <li>第三项</li>
  </ul>
  <hr>
  有顺序列表:
  <ol>
    <li>第一项</li>
    <li>第二项</li>
    <li>第三项</li>
  </ol>
</body>
</html>

<!-- 设置网页头标题-->
<!-- 主体标记的开始-->
<!-- 使用段落标记输出一个文本-->
<!--居中设置一段文本-->
<!-- 预定义标记保留源代码格式输出 -->
<!-- 保留这此文字在源代码中的格式 -->
<!-- 保留这此文字在源代码中的格式 -->
<!-- 保留这此文字在源代码中的格式 -->
<!-- 保留这此文字在源代码中的格式 -->
<!-- 在段落与段段落之间输出的分隔 -->
<!-- 在页面中输出一个文本 -->
<!-- 无顺序列表的开始标记 -->
<!-- 无顺列表中第一个列表项目 -->
<!-- 无顺列表中第二个列表项目 -->
<!-- 无顺列表中第三个列表项目 -->
<!-- 无顺序列表的结束标记 -->
<!-- 在段落与段段落之间输出的分隔 -->
<!-- 有顺序列表的开始标记 -->
<!-- 有顺序列表的开始标记 -->
<!-- 有顺列表中第一个列表项目 -->
<!-- 有顺列表中第二个列表项目 -->
<!-- 有顺列表中第三个列表项目 -->
<!-- 有顺序列表的结束标记 -->
<!-- 主体标记的结束标记 -->
<!-- HTML文件结束 -->
```



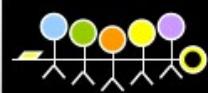


4.2 文本标签

标签	描述
<hn>	标题字标签，n为1-6，定义六级标题，而且会自动换行插入一个空行
	粗体字标签
<i>	斜体字标签
<u>	下划线字体标签
<sub>	文字下标字体标签
<sup>	文字上标字体标签
	字体标签，可以通过标签的属性指定文字的大小、颜色及字体等信息
<tt>	打字机文字
<cite>	用于引证、举例、通常为斜体字
	表示强调，通常为斜体字
	表示强调，通常为粗体字
<small>	小型字体标签
<big>	大型字体标签



```
<html>
<head><title>文本标记测试网页</title></head>
<body>
  <h1>使用<h1>H1</h1>; 输出一号标题字体</h1>
  <h2>使用<h2>H2</h2>; 输出二号标题字体</h2>
  <h3>使用<h3>H3</h3>; 输出三号标题字体</h3>
  <h4>使用<h4>H4</h4>; 输出四号标题字体</h4>
  <h5>使用<h5>H5</h5>; 输出五号标题字体</h5>
  <h6>使用<h6>H6</h6>; 输出六号标题字体</h6>
  <font face="楷体_GB2312" color="red" size="5">绝对字体大小为5的红色楷体字
</font><br>
  <font face="宋体" color="green" size="+3">相对字体大小为+3的绿色宋体字
</font><br>
  <font face="黑体" color="blue" size="-1">相对字体大小为-1的蓝色黑体字
</font><br>
  <b>使用<b>B</b>; 标记输出粗体字</b><br>
  <u>使用<u>U</u>; 标记输出带有下划线的文字</u><br>
  <i>使用<i>I</i>; 标记输出斜体字</i><br>
  <tt>使用<tt>TT</tt>; 标记输出打印机文字</tt><br>
  <cite>使用<cite>CITE</cite>; 标记输出引证、举例的斜体字</cite><br>
  <em>使用<em>EM</em>; 标记输出强调的斜体字</em><br>
  <strong>使用<strong>STRONG</strong>; 标记输出强调的粗体字</strong><br>
  <small>使用<small>SMALL</small>; 标记输出小型的字体</small><br>
  <big>使用<big>BIG</big>; 标记输出大型的字体</big><br>
</body> </html>
```



Microsoft Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

后退 前进 刷新 搜索 收藏夹

地址(A) G:\易第优教育\细说php\《细说PHP》配套源码\Chapter_4\test2.html 转到 链接

使用<H1>输出一号标题字体

使用<H2>输出二号标题字体

使用<H3>输出三号标题字体

使用<H4>输出四号标题字体

使用<H5>输出五号标题字体

使用<H6>输出六号标题字体

绝对字体大小为5的红色楷体字

相对字体大小为+3的绿色宋体字

相对字体大小为-1的蓝色黑体字

使用标记输出粗体字

使用<U>标记输出带有下划线的文字

使用<I>标记输出斜体字

使用<TT>标记输出打印机文字

使用<CITE>标记输出引证、举例的斜体字

使用标记输出强调的斜体字

使用标记输出强调的粗体字

使用<SMALL>标记输出小型的字体

使用<BIG>标记输出大型的字体

完毕 我的电脑



五、创建图像和链接

- ❖ 5.1 插入图片
- ❖ 5.2 建立锚点和超链接

5.1 插入图片

❖ **img**标签 -- 代表HTML图像

- **img**标签是单独出现的: ``, **img**是**image**(图像)的缩写
- 常用属性:
 - **alt** -- 代表图像的替代文字
 - **src** -- 代表一个图像源(就是图像的位置)
 - **border** -- 代表图片边框的宽度
 - **height** -- 代表一个图像的高度
 - **width** -- 代表一个图像的宽度
- 设计网页时经常使用的图片有三种格式:
 - **GIF** -- 最多支持256色,支持透明,支持多帧动画显示效果.
 - **JPEG** -- 支持多种颜色,可以有很高的压缩比,使用了有损压缩,不支持透明,不支持动画效果.
 - **PNG** -- 是一种新的图片技术,可以表现品质比较高的图片,使用了无损压缩,支持透明,不支持动画.



图像的位置

为图像添加的提示性文字

`` `<!-- 使用alt属性的图片 -->`

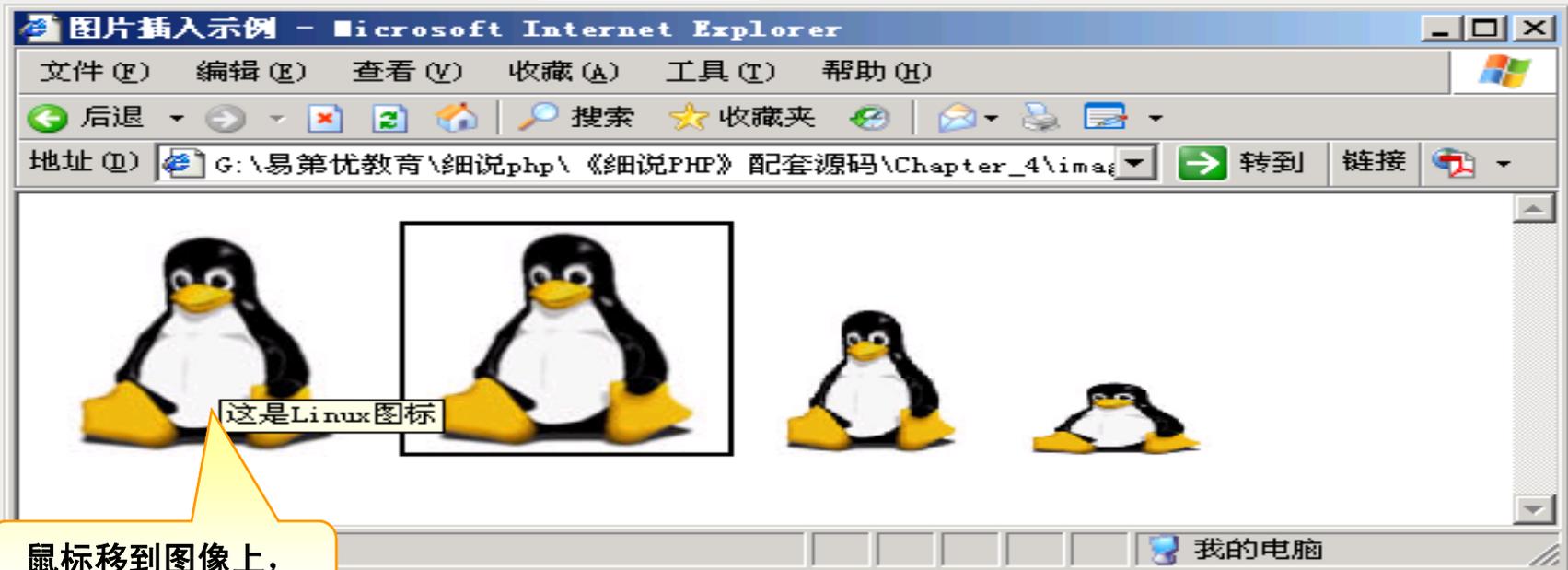
`` `<!-- 设置图片的边框宽度2px -->`

`` `<!-- 只设置宽度高度自动适应 -->`

`` `<!-- 图片高度和宽度一起设置 -->`

图像的宽度

图像的高度



鼠标移到图像上，
出现的提示性文字



❖ map标签 -- 定义图像点击区域的集合

- map标签是成对出现的, 即`<map>...</map>`, 而且必须配合area标签使用。
- map标签中name与id属性指定的值必须与定义图像点击区中图像(img)的usemap属性指定的值一致。
- 属性
 - id -- 定义map的名称。
 - name -- 定义map的名称。

❖ area标签 -- 定义图像点击区域,

- area标签是单独出现的, `<area />`
- area标签必须使用在map标签中, 而且必须配合img标签使用。
- 属性
 - accesskey -- 链接的快捷键访问方式
 - alt -- 图像的提示文字
 - coords -- 定义可点击区域图形的坐标
 - href -- HTML链接源的URL
 - nohref -- 图像点击排除的区域, 当不使用href时应使用nohref
 - shape -- 可点击区域的形状
 - tabindex -- 使用"Tab"键的遍历顺序
 - target -- 链接目标

❖ <map>与<area>标签应用实例

```

```

```
<map name="Map" id="Map">
```

```
<area shape="rect" coords="35,29,135,99" href="#" />
```

```
<area shape="circle" coords="243,78,44" href="#" />
```

```
<area shape="poly" coords="120,137,195,154,135,207" href="#" />
```

```
</map>
```

rect定义一个矩形区域，**coords**属性设置值为矩形的左上角、右下角的坐标

poly定义一个多边形区域，**coords**属性设置值为多边形各顶点的坐标值

circle定义一个圆形区域，**coords**属性设置值分别为圆心的横、纵坐标，第三个参数为半径



5.2 建立锚点和超链接

- ❖ **a标签** -- 代表HTML链接
- ❖ a, 是anchor (锚) 的第一个字母, a标签是成对出现的, 即<a>...。
- ❖ 常用属性:
 - href -- 代表一个url链接源 (就是链接到什么地方)
 - url除了是网页外, 还可以是其它的文件 (如文本文件, pdf文件等)。
 - url还可以是指向HTML文件中的一个位置。
 - url还可以是Email地址。
 - target -- 用来指出哪个窗口或框架应该被此链接打开
 - target=_blank: 将链接内容在新的浏览窗口中打开。
 - target=_parent: 将链接的内容, 当成文件的上一个画面。
 - target=_self: 将链接的内容, 显示在目前的窗口中。
 - target=_top: 这个参数可以解决新内容被旧框窗包围的困扰, 使用这参数, 会将整个画面重新显示成链接的画面内容。
 - title -- 代表链接的附加提示信息
- ❖ 链接语法:

```
<a href="url" target="_blank">显示的文字</a>
```



❖ 定位网页内部的链接（锚点链接）：

- 用 `` 定义，例如：`第一部分`，使用 `` 跳转到第一部分 `` 超链接就可以定位到网页中的“第一部分”这个位置。

❖ 注意：`...` 与 `...` 的区别

❖ URL (Uniform Resource Locator) 的基本组成：

- 协议名：`//` 主机名：`端口号/资源名`。
- 例如：`http://www.lampuser.com:8080/index.html`

❖ 为URL指定参数

- 例如：
`http://www.lampuser.com/register.html?name=zhangsan&password=123`

❖ 带有定位标记的URL

- 例如：
`http://www.lampuser.com/index.html#section2`



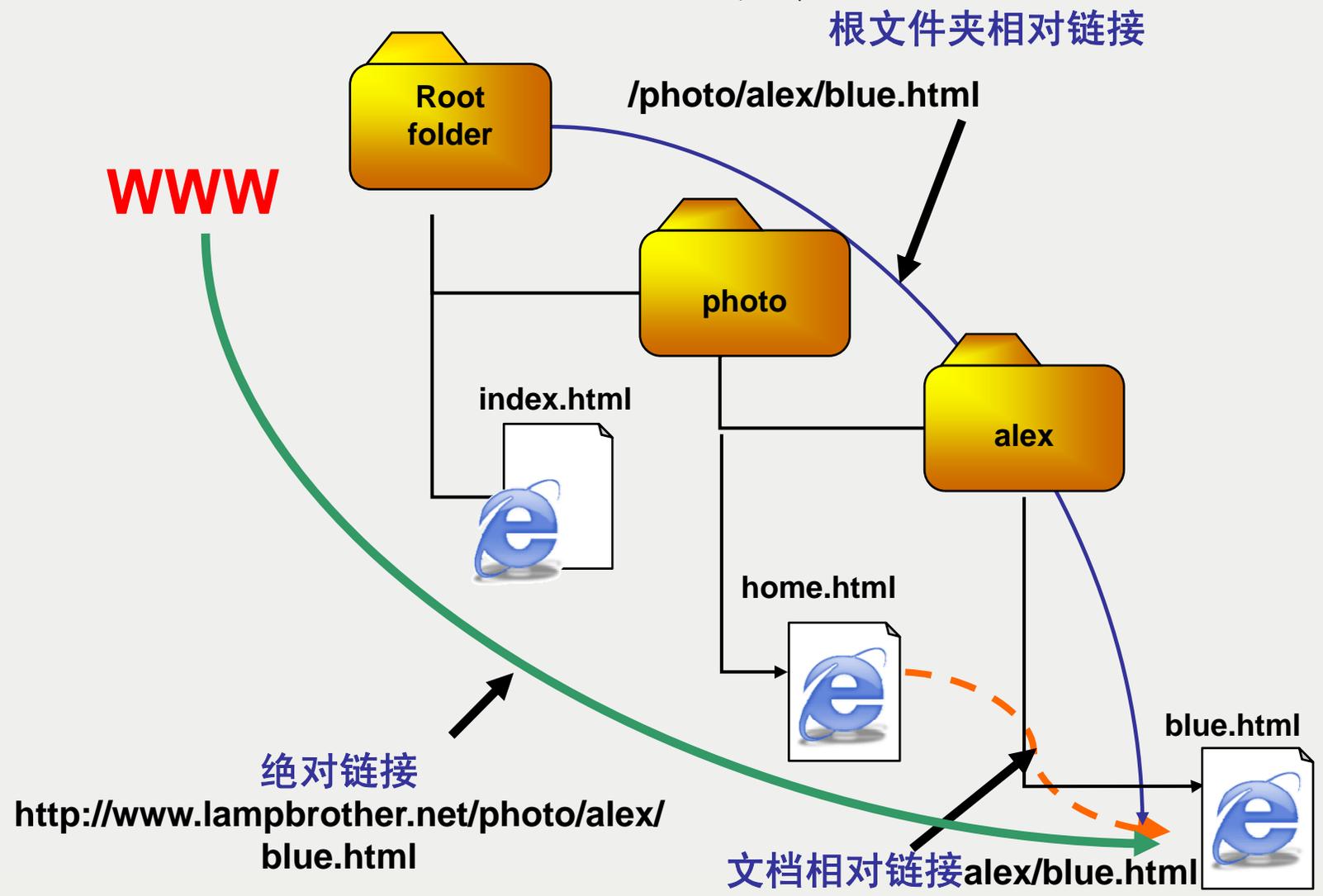
❖ 超链接范例:

- 网站链接: `PHP专家`
- 电子邮件链接: `写信给我`
- ftp链接: `下载档案`

❖ 有三种类型的链接路径:

- **绝对路径**
(例如 `http://www.sohu.com/index.html`)
- **文档相对路径**
(例如 `adver/contents.html`)
- **站点根目录相对路径**
(例如 `/support/app/customer.html`, 其中 “/” 表示根目录)

链接路径





URL 编码

❖ 规则:

- 将空格转换为加号 (+)。
- 对 0-9, a-z, A-Z 之间的字符保持不变。
- 对于所有其他的字符, 用这个字符的当前字符集编码在内存中的十六进制格式表示, 并在每个字节前加上一个百分号 (%), 如字符 “+” 用 %2B 表示, 字符 “=” 用 %3D 表示, 字符 “&” 用 %26 表示, 第个中文字符在内存中占两个字节, 字符 “中” 用 %D6%D0 表示, 字符 “国” 用 %B9%FA 表示。
- 对于空格也可以直接使用其十六进制编码方式, 即用 %20 表示, 而不是将它转换成加 (+)。



思考与实践

- ❖ 描述什么是HTML标签及其作用
- ❖ 描述DTD文件的作用及其与HTML文件的关系。
- ❖ 描述什么是URL及其组成部分，举一个具有端口号的URL例子。
- ❖ 编写一个HTML网页文件，在网页文件中加入5个大的段落，在文件开始处是每个段落 的标题名，单击任何一个标题名，浏览器定位到该标题对应的段落处显示。



六、使用XHTML表格

- ❖ 6.1 表格标签常见标签
- ❖ 6.2 表格标签常见属性
- ❖ 6.3 综合示例



6.1 表格中常用标签

标签	描述
<table>	代表HTML表格，table是成对出现的，以<table>开始，以</table>结束
<caption>	用于定义表格标题的位置，可由align和valign属性来设置。
<tr>	代表HTML表中的一行，此标签也要求成对出现。
<th>	代表HTML表格中的表头，一般位于表格的首行。要求成对。
<td>	代表HTML表格中的一个单元格。用于存放具体数据内容。

❖ 表格的基本结构：

编号	列	姓名	年龄	备注
1	...	小王	20	
2	...	小李	21	
...

表格的基本语法

border用来设置表格边框尺寸大小

```
<table border="1">  
  <tr>  
    <td>  
      单元格内容  
    </td>  
    .....  
  </tr>  
  .....  
</table>
```

<td>...</td>
定义列

<tr>...</tr>
定义行

<table>...</table>
定义表格



6.2 表格标签常见属性

❖ table标签属性:

- **width** -- 代表表格的宽度
- **height** -- 代表表格的高度
- **border** -- 代表表格边框(此属性应该使用CSS实现)
- **cellspacing** -- 代表表格边框与表格内容填充的距离,也是内容填充之间的距离(此属性应该使用CSS实现)
- **cellpadding** -- 代表内容填充的宽度(此属性应该使用CSS实现)

❖ th、td标签属性:

- **width与height** - 代表宽度与高度
- **colspan** -- 一行跨越多列
- **rowspan** -- 一列跨越多行
- **align** -- 代表水平对齐方式(left(左对齐) | center(居中对齐) | right(右对齐) | justify)(此属性应该使用CSS实现)
- **valign** -- 代表垂直对齐方式(top(顶部对齐) | middle(中部对齐) | bottom(下部对齐) | baseline(基线对齐))(此属性应该使用CSS实现)

❖ tr标签属性:

- **align与valign**具体解释详见上面。

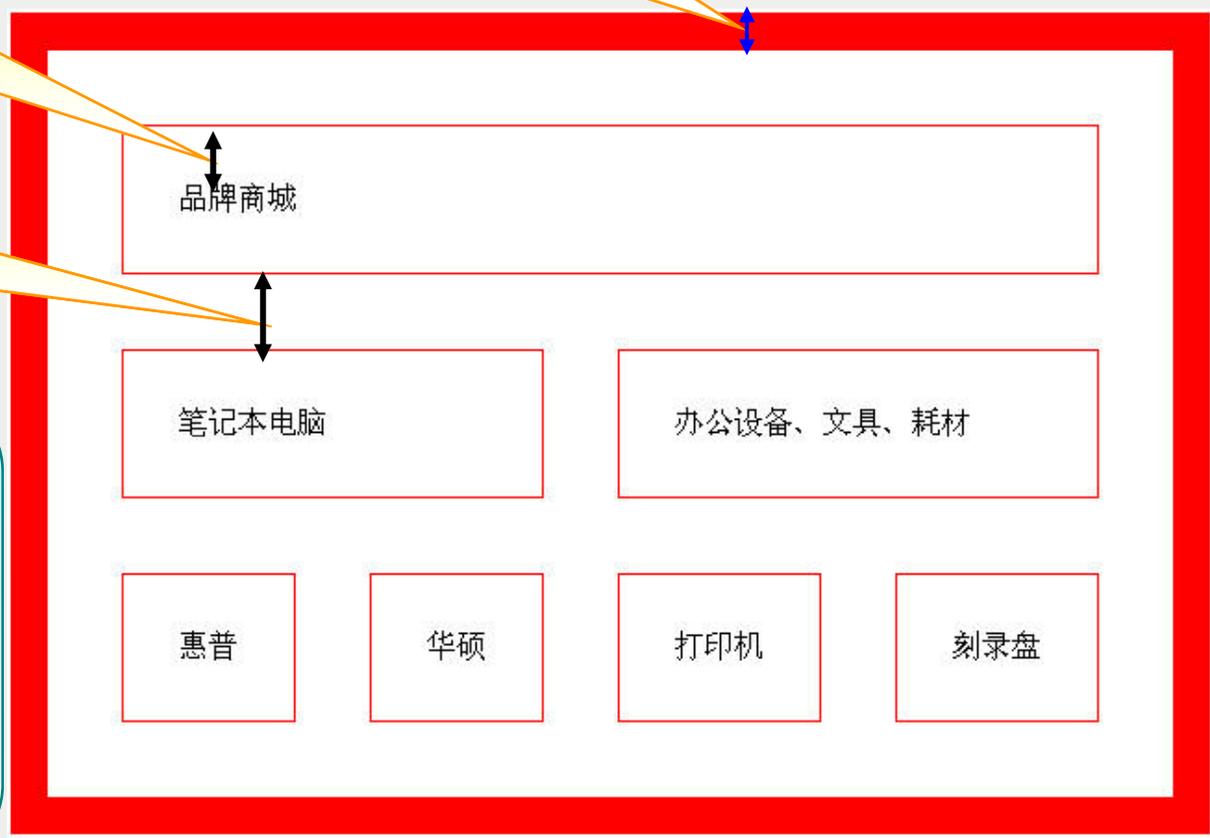
如何使用填充、间距属性

border(边框的厚度)

cellpadding
(单元格填充)

cellspacing
(单元格间距)

```
<table border="20" cellpadding="30" cellspacing="40" bordercolor="red">
.....
</table>
```





6.3 综合示例

```
<table width="80%" border="1"
  cellpadding="3" cellspacing="0"
  align="center">
  <caption><h1>学员表</h1></caption>
  <tr>
    <th colspan="3">学员基本信息</th>
    <th colspan="2">成绩</th></tr>
  <tr>
    <th>姓名</th><th>性别</th>
    <th>专业</th><th>课程</th>
    <th>分数</th>
  </tr><tr align="center">
    <td>小峰</td><td>男</td>
    <td rowspan="2">计算机</td>
    <td rowspan="2">PHP开发</td>
    <td>86</td></tr>
  <tr align="center">
    <td>小影</td>
    <td>女</td><td>98</td>
  </tr>
</table>
```

学员基本信息			成绩	
姓名	性别	专业	课程	分数
小峰	男	计算机	PHP开发	86
小影	女			98



小结

- ❖ 创建表格最少需要那三个标签？
- ❖ 简述表格的基本结构。
- ❖ 跨行跨列的表格，主要在什么情况下使用？
- ❖ 给你一个表格，你会从哪些方面进行美化？



七、XHTML框架结构

- ❖ 7.1 HTML框架简述
- ❖ 7.2 HTML框架标签
- ❖ 7.3 HTML框架示例



7.1 HTML框架简述

❖ HTML框架简述

- 一个浏览器窗体可以通过几个页面的组合来显示。我们可以使用框架来完成(frames)这项工作。(框架可以把HTML文档分为多个页面)。也就是将一个浏览器文档窗口分隔成多个窗口，每个窗口都可以显示一个独立的网页文件。
- 框架页使用了表格的方式组合，可以分为数行与数列。

❖ 框架的优点

- 重载页面时不需要重载整个页面，只需要重载页面中的一个框架页(减少了数据的传输，增加了网页下载速度)。
- 方便制作导航栏。

❖ 框架的缺点

- 会产生很多页面，不容易管理。
- 不容易打印。
- 浏览器的后退按钮无效。
- 代码复杂，无法被一些搜索引擎索引到。
- 多数小型的移动设备(PDA 手机)无法完全显示框架。
- 多框架的页面会增加服务器的http请求。
- 由于上面诸多缺点，因此不符合标准网页设计的理念，已经被标准网页设计抛弃。



7.2 HTML框架标签

- ❖ **frameset** 标签 -- 代替**body**标签定义了框架页, 并且定义了框架将分为多少行与多少列
 - 常用属性:
 - **cols** -- 定义了框架含有多少列与列的大小 (每个值使用逗号分隔), 取值为像素**px**或者百分比%
 - **rows** -- 定义了框架含有多少行与行的大小 (每个值使用逗号分隔), 取值为像素**px**或者百分比%
 - **border** -- 定义frame定义的框架页的边框 (单位像素), 使用**css**实现
 - **frameborder** -- 定义框架页是否边框 (此属性应写在**frame**标签内部, 不应在此出现)
 - **framespacing** -- 定义框架页之间间隔的距离, 使用**css**实现
- ❖ **<noframes>**标签
 - 可为那些不支持框架的浏览器显示文本, 和**<body>**组合使用
- ❖ **<iframe>**标签
 - 创建一个包含另外一个文档的内联框架, **iframe**标签内的内容可以做为浏览器不支持**iframe**标签时显示。



❖ frame标签 -- 定义frameset标签中每个框架页的内容

- frame标签是单独出现的, `<frame />`
- 常用属性:
 - **frameborder** -- 定义了内容页的边框, 取值为 `(1|0)`, 缺省值为 `1`
 - `1` -- 在每个页面之间都显示边框
 - `0` -- 不显示边框
 - **name** -- 在一个框架页链接到另一框架页时使用 (另一个框架页可以使用 **target** 定义链接页)
 - **noresize** -- 定义了浏览者是否可以通过拖拽改变框架页尺寸, 取值为 `(noresize)`
 - **scrolling** -- 定义是否有滚动条, 取值为 `(yes|no|auto)`, 缺省值为 `auto`
 - `yes` -- 显示滚动条
 - `no` -- 不显示滚动条
 - `auto` -- 当需要时再显示滚动条
 - **src** -- 定义了内容页URL
 - **border** -- 设置边框粗细



7.3 HTML框架示例

```
<html>                                <!-- 网页开始标记 -->
  <head>
    <title>使用框架定义后台管理平台模型</title>
  </head>                                <!-- 设置网页标题 -->
  <frameset rows="80,*" frameborder="1" border="5"> <!-- 划分两行 -->
    <frame src="top.html" name="top" scrolling="no" noresize/>
                                          <!-- 顶部大类窗体 -->
    <frameset cols="200,*">              <!-- 划分左右两列 -->
      <frame src="menu.html" name="menu" scrolling="no" noresize/>
                                          <!-- 左边菜单窗体 -->
      <frame src="main.html" name="main" noresize />
                                          <!-- 右边内容窗体 -->
    </frameset>                          <!-- 内层框架结束 -->
  </frameset>
  <noframes>
    <body><p>您的浏览器不支持框架，请升级浏览器</p></body>
  </noframes>
</frameset>                             <!-- 外层框架结束 -->
</html>
```



框架集窗口中的各个页面

Top.html文件:

```
<center>这是顶部的网页文件</center>
```

Left.html文件:

```
<p>这是左边的网页文件</p>
```

```
<p><a href="right1.html" target="right">显示右边的第一个文档</a>
```

```
</p>
```

```
<a href="right2.html" target="right">显示右边的第二个文档</a>
```

Right1.html文件:

这是右边的第一个网页文件

Right2.html文件:
这是右边的第二个网页文件



换名登陆

查看首页

退出系统

在线帮助

当前用户:

当前位置: 网站首页

数据表管理

添加数据表

查看数据表

数据管理

信息审核

数据备份

系统管理

暂无管理

欢迎您用户登陆本网站

今天是: 2008年4月15日星期二

这是您: 第 次登陆本网站

您在本网站拥有权限:

拥有权限等级:

拥有访问权限:

数据添加权限:

数据编辑权限:

数据审核权限:

用户密码维护:



八、XHTML表单设计格式标签

- ❖ 8.1 HTML form标签
- ❖ 8.2 HTML input标签
- ❖ 8.3 HTML textarea标签
- ❖ 8.4 HTML select标签



8.1 HTML form标签

- ❖ 问：网站怎样与用户进行交互？ 答案：使用HTML表单(form)。
- ❖ 表单是可以把浏览者输入的数据传送到服务器端的程序(比如ASP, [PHP](#))的 [HTML元素](#)，服务器端程序可以处理表单传过来的数据，从而进行一些动作。比如，bbs, blog的登陆系统, 购物车系统等。
- ❖ **form** 标签 -- 代表HTML表单
 - **form**标签是成对出现的, 以<form>开始, 以</form>结束
 - 常用属性.
 - **action** -- 浏览者输入的数据被传送到到的地方, 比如一个PHP页面 (dreamdu.php)
 - **method** -- 数据传送的方法
 - **get** -- 此方式传递数据量少, 但是传递的信息显示在网址上。
 - **post** -- 此方式传送信息多, 而且不会把传递信息显示在网址上
 - **enctype** -- 表示将数据发送到服务器时浏览器使用的编码类型
 - **application/x-www-form-urlencoded** -- 窗体数据被编码为名称/值对. 这是标准的编码格式. 默认的。
 - **multipart/form-data** -- 窗体数据被编码为一条消息, 页上的每个控件对应消息中的一个部分。
 - **text/plain** -- 以纯文本形式进行编码, 其中不含任何控件或格式字符。



8.2 HTML input 标签

❖ input 标签 -- 代表HTML表单的单行输入域

- input 标签是单独出现的, `<input />`

- 属性.

- **type** -- 代表一个输入域的显示方式 (分为输入型, 选择型, 点击型)
- **name** - 此表单项名称
- **value** -- 输入域的值
- **size** -- 输入域的长度
- **maxlength** -- 输入域最多可以输入文字的长度
- **checked** -- 如果是选择型的输入域, 代表已经被选择, 值为checked
- **readonly** -- 输入域可以选择, 但是无法修改, 值为readonly
- **disabled** -- 输入域无法获得焦点, 无法选择, 以灰色显示, 在表单中不起任何作用。如: `disabled="disabled"`
- **accesskey** -- 表单的快捷键访问方式, 如值为h即按Alt+h快捷键。
- **tabindex** -- 输入域的"tab"键遍历顺序
- **src** -- 当使用图片来表示按钮时, 代表图片的位置 (URI)
- **alt** -- 用来替换提交按钮的图片 (当在input的src属性定义的图片无法显示时) 提示信息。



- ❖ **type**属性 -- 代表HTML表单, 单行输入域(框)的表现方式
- ❖ **type**属性取值:
 - **text** -- 文字输入域(输入型)
 - **password** -- 也是文字输入域, 但是输入的文字以密码符号 '*' 显示(输入型)
 - **file** -- 可以输入一个文件路径(输入型)
 - **checkbox** -- 复选框. 可以选择零个或多个(选择型)
 - **radio** -- 单选框. 只可以选择一个而且必须选择一个(选择型)
 - **hidden** -- 代表隐藏域, 可以传送一些隐藏的信息到服务器
 - **button** -- 按钮(点击型)
 - **image** -- 使用图片来显示按钮, 使用src属性指定图像的位置(就像img标签的src属性)(点击型)
 - **submit** -- 提交按钮, 表单填写完毕可以提交, 把信息传送到服务器. 可以使用value属性来显示按钮上的文字(点击型)
 - **reset** -- 重置按钮, 可以把表单中的信息清空(点击型)



8.3 HTML textarea 标签

- ❖ **textarea 标签** -- 代表HTML表单多行输入域
 - 此标签是成对出现的,即以<textarea>开始,以</textarea>结束
 - 属性:
 - **cols** -- 多行输入域的列数
 - **rows** -- 多行输入域的行数
 - **name** -- 此表单项名称
 - **accesskey** -- 表单的快捷键访问方式
 - **disabled** 无法获得焦点,无法选择,灰色显示,表单中无效
 - **readonly** 输入域可以选择,但是无法修改
 - **tabindex** 输入域,使用"tab"键的遍历顺序

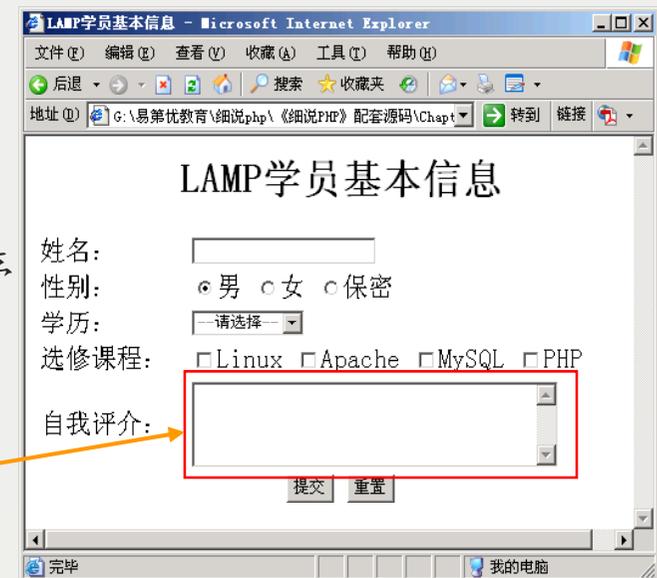
```
<td>自我简介: </td>
```

```
<td>
```

```
<textarea name="eval" rows="4" cols="40">
```

```
</textarea>
```

```
</td>
```





8.4 HTML select 标签

❖ select 标签 -- 选择列表标签

- select 标签是成对出现的, 以<select>开始, 以</select>结束
- 此标签中的每对option 标签代表一个选择项
- 属性:
 - name - 表单项名称
 - size -- 选择域的高度
 - multiple -- 可以有多个选择
 - disabled -- 以灰色显示, 在表单中不起任何作用
 - tabindex -- 使用"tab"键的遍历顺序

❖ option 标签 -- 代表选择列表的一个选择项

- option 标签是成对出现的, 以<option>开始, 以</option>结束
- 属性:
 - label -- 说明选择项
 - value -- 说明选择项的值
 - selected -- 此选择项已经被选择
 - disabled -- 输入框无法获得焦点, 无法选择, 以灰色显示, 在表单中不起任何作用
 - tabindex -- 使用"tab"键的遍历顺序



表单包含的控件

单行文本输入框(text)

单选按钮(radio)

复选框(checkbox)

重置按钮(reset)

多行文本框(textarea)

密码框(password)

下拉列表(select)

提交按钮(submit)

淘宝网注册页 - Microsoft Internet Explorer

文件(E) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

地址(D) D:\使用HTML语言开发商业站点\HTML_chapter3\PPT\chap

登录名: (可包含 a-z、0-9 和下划线)

密码: (至少包含 6 个字符)

再次输入密码:

性别: 男 女

爱好: 运动 聊天 玩游戏

出生日期: 年 [选择月份] 月 日

阅读淘宝网服务协议

本服务协议双方为淘宝与淘宝网用户, 本服务协议具有合同效力。
淘宝的权利和义务
1. 淘宝网有义务在现有技术维护整个网上交易平台的正常运行, 并努力提升和改进技术, 使用户网上交易活动的顺利。
2. 对用户注册使用淘宝网中遇到的与交易或注册有关的问题及反映的情况, 淘宝应及时作出回复。

完毕 我的电脑



小结

- ❖ 表单中提交数据的方法有那两种及其区别？
- ❖ 表单里有哪些常用的表单元素？
- ❖ 创建一个框架页面至少需要哪两个标签？
- ❖ target设定目标资源所要显示的窗口，其取值可以为哪些？
- ❖ 简述创建一个多框架页面所需的主要步骤。