



# 软件开发环境

Software Development Environment

主讲教师 刘凡  
[fanliu@hhu.edu.cn](mailto:fanliu@hhu.edu.cn)



# 第二章

## Web开发技术概述

-HTML、CSS、DHTML、XML、J2EE



## 1、HTML概述



Web是一种典型的分布式应用架构。

Web应用中的每一次信息交换都涉

及到客户端和服务端两个层面。客

户端的主要任务是展现信息内容，而HTML语言则是信

息展现的最有效载体之一。因此，HTML是万维网的基

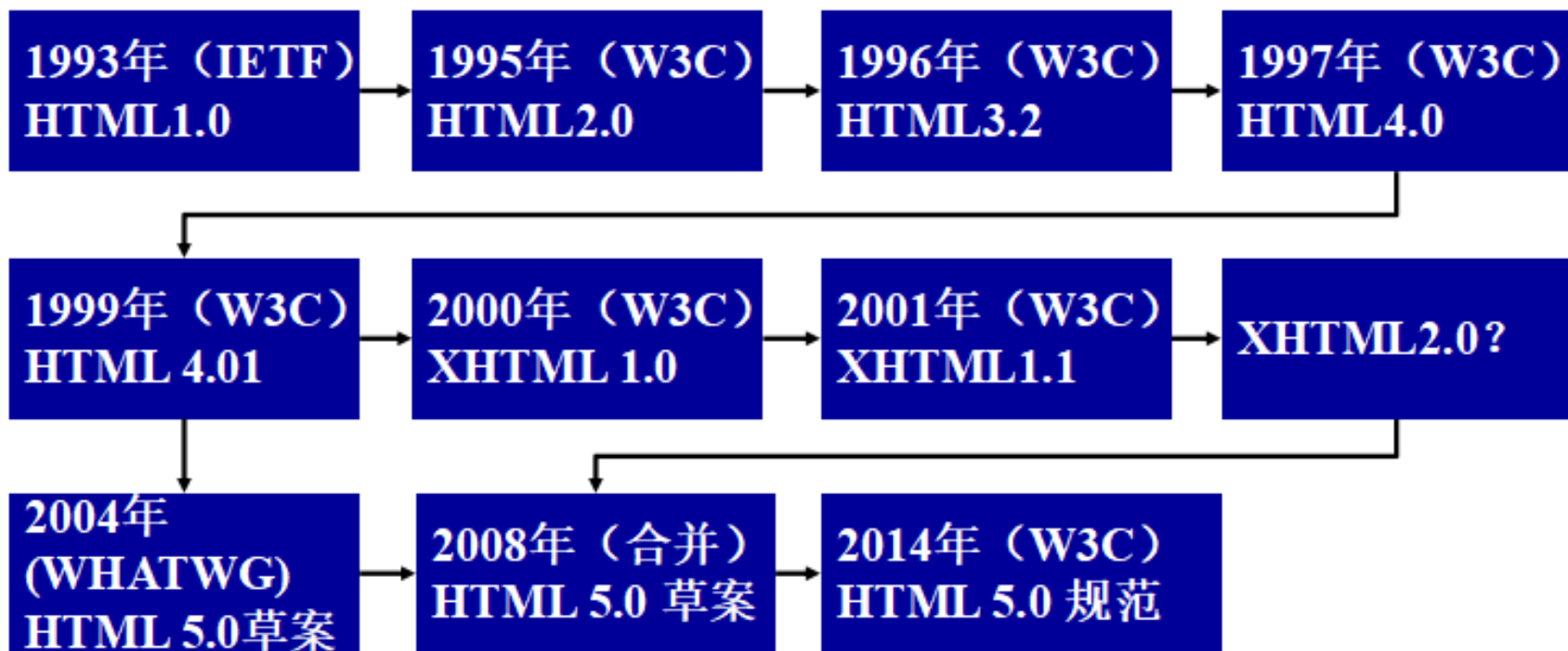
础技术之一。HTML是用于描述网页文档的标记语言。

常用数字来描述HTML的版本。





## 2、HTML发展史





## HTML5与原生APP对比








VS



- 👉 在跨平台上，HTML5胜出
- 👉 在用户体验和表现上原生APP开发胜出
- 👉 在开发速度和成本方面，HTML5胜出
- 👉 在版本控制和BUG修复方面，HTML5胜出
- 👉 在变现能力方面，原生APP开发胜出
- 👉 对于碎片化挑战而言，HTML5胜出

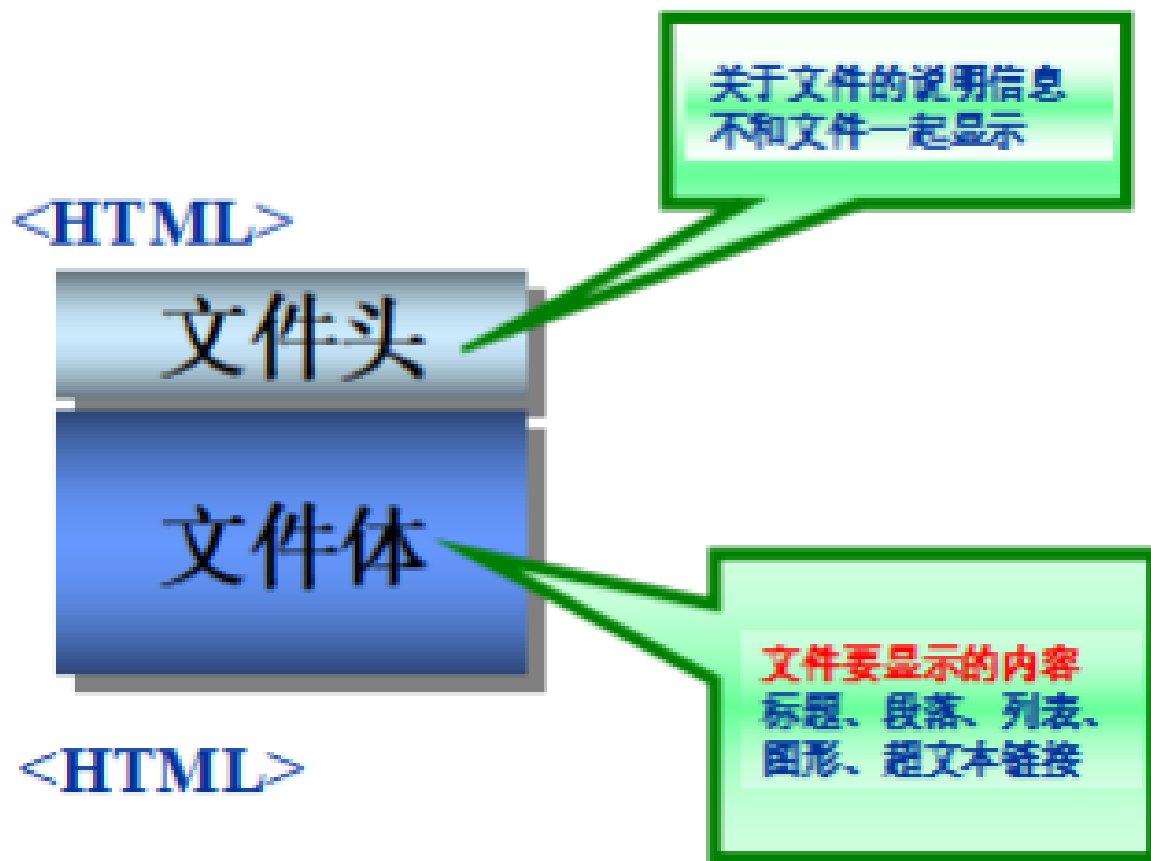


## HTML超文本标记语言特征:

-  格式化的标记语言
-  文档 = 控制标记 + 显示内容
-  简易性
-  可拓展性
-  与平台无关



## HTML文件的结构





# HTML







<html>	<!--<html>文档开始-- >
<head>	<!--说明页面属性的文件头开始 -- >
<title>一个简单的html文件</title>	<!--页面标题 -- >
</head>	<!--文件头的结束-- >
<body>	<!--文件体开始，文件体中内容在浏览器中显示 -- >
欢迎！	
请单击<a href="ex05-001_1.html">学习web技术！ </a>	
</body>	<!--文件体结束-- >
</html>	<!-- <html>文档的开始-- >



## 制作步骤:

- (1) 使用简单的文字编辑器，如Notepad(记事本)、WordPad(书写板)等都可以。在编辑器中输入两个HTML文件的代码。
- (2) 以扩展名 “.html”保存文件。在只支持三个字母作后缀的操作系统中,扩展名是 “.htm”。
- (3) 在浏览器地址栏目输入文件名,显示页面。




# HTML页面修饰标记

## 1、HTML的标记和元素

 控制语句称为**标记**，格式如下：

**< 标记 > HTML语言元素 < / 标记 >**

如：<H2> 你好！ </H2>

 标记大多成对出现，由<始标记>和</尾标记>组成。

始标记通知浏览器开始执行该标记指示的功能，

尾标记说明该功能到此结束。



# HTML页面修饰标记

## 1、HTML的标记和元素

- ✎ 标记与大小写无关
- ✎ 标记可以联合使用，也可以嵌套使用
- ✎ 标记可以带有一个或多个属性参数，格式为：  
**<标记 属性1=“属性值1” 属性2=“属性值2” ...>**
- ✎ 注释标记**<!--注释内容-->**不在浏览器中显示，只供阅读页面代码帮助理解使用。



## 2、页面结构标记



### **<html>标记**

在html文档最外层，表示html文件开始与结束。



### **<head>文件头标记**

在文件头标记<head>与</head>之间保存说明整个文件的综合信息。包含的标记：

#### **<title>标题标记**

**<title > 标题 </title>**



# HTML页面修饰标记

## 2、页面结构标记



### **<body>**文件体标记

标明文档主体，是HTML文档主要部分。**<body>**表示文档的开始，**</body>**表示文档的结束。**<body>**标记可以使用属性使页面带有背景颜色或背景图案，主要属性如下：

**background:** 背景图案或图像文件的URL

**bgcolor:** 背景颜色

6位16进制的红-绿-蓝 (red-green-blue, RGB) 值表示  
#ff0000—红色 ， #00ff00—绿色， #0000ff—蓝色



## 3、页面修饰标记

控制页面的段落，显示字符的大小、颜色、字体和属性。



### 字体标记

定义了六级标题，`<h1>`、`<h2>`、.....、`<h6>`，`<h1>`字体最大最黑



**`<font>`**标记描述字体的颜色、大小和字体，使用属性：

**color:** 定义字体颜色，`<font color="red">`

**face:** 定义字体类型，`<font face="宋体">`

**size:** 字体大小，`<font size="n">`，n从1到7，7号最大



# HTML页面修饰标记

## 3、页面修饰标记



### 段落标记

HTML文档中的空格、Tab符、回车换行符等，在浏览器中不起作用，必须使用标记，使文章分出段落，显出层次。

**<p>分段标记**：表示新一段开始，段落间有一空行

**<br>换行标记**：另起一行，中间不插入空行

**<hr>水平线标记**：在页面上画出一条水平线

**<hr color=“#rrggbb” size=“n”>**：宽度为n像素直线

**<pre>预格式化标记**：使HTML文档中的空格、Tab符、回车换行符起作用，与尾标记</pre>一起使用





# HTML页面修饰标记

## 3、页面修饰标记



### 文字对齐标记

`<center> ... </center>`

文字居中

`<left> ... </left>`

文字左对齐

`<right>...</right>`

文字右对齐格式

这些标记可以复合使用



---



# HTML页面修饰标记的应用

显示上标: `x<sup>3</sup>`

显示下标: `x<sub>2</sub><br>`

`<hr>`

`<h2 align="center" >显示2号字</h2></center>`

`<pre>`

\*

\* \*

\*

`</pre>`

`</body>`

`</html>`



# HTML页面修饰标记的应用





# 页面多媒体技术

## 网页多媒体技术



### <img>图像标记

把图像嵌入html文档，属性：

**src:** 必选项，指定图像文件的url

**alt:** 定义一个文本串，浏览器尚未完全读入图像或因故不能显示图像时，在图像位置使用文本串代替图像的显示。

当浏览器可以显示图像时，alt属性不起作用

**align:** 文本与图像的对齐方式，取值：left、middle、right、top和bottom，align=bottom表示图像底部与文本对齐

**border:** 图像边框的宽度(以像素为单位)，如 border=1，border=0取消边框

**width、height:** 定义图像高和宽的像点数目



# 页面多媒体技术

## 网页多媒体技术



### <bgsound> 背景音乐标记

在展示页面的同时播放音乐， 属性：

**src:** 音乐文件的url

**loop:** 音乐的播放次数， loop=-1 音乐不断循环播放

如： <bgsound src="9.mid" loop="-1">



# 页面多媒体技术的应用

---

<html>

<head><title>图像标记的应用</title>

<bgsound src="9.mid" loop="-1">

</head>

<body>



<br>圣诞快乐！

</body>

</html>



# 表格与列表标记

## 1、表格标记



**<table>** 表格标记： 建立表格

**<table>**与**</table>**之间为表格标题、表头及单元格中的内容

**border**属性： 表格边框宽度，以像素为单位



**<caption>**与**</caption>**： 定义表格标题

**align**属性： **align=top**， **align=bottom**



**<tr>**与**</tr>**： 定义表格中的一行



**<th>**与**</th>**： 定义表头元素，表头显示成黑体



**<td>**与**</td>**： 定义单元格内容





# 表格与列表标记的应用

## 例：课表代码清单

```
<html>
<head><title>表格标记应用</title>
</head>
<body>
<table border>
<caption>课表</caption>
<tr>
<th>节次</th><th>星期一</th><th>星期二</th><th>星期三</th>
<th>星期四</th><th>星期五</th>
</tr>
<tr>
<td>1、 2</td><td>基础英语</td><td>操作系统</td><td>网络基础
</td>
<td>基础英语</td><td>数据库</td>
</tr>
```



# 表格与列表标记的应用

```
<tr>
<td>3、 4</td><td>Java</td><td>数据库</td><td>实验</td>
<td>Java</td><td>操作系统</td>
</tr>
<tr>
<td>5、 6</td><td>网络基础</td><td>实验</td><td>实验</td>
<td>实验</td>
</tr>
</table>
</body>
</html>
```



# 表格与列表标记的应用

课表					
节次	星期一	星期二	星期三	星期四	星期五
1、2	基础英语	操作系统	网络基础	基础英语	数据库
3、4	Java	数据库	实验	Java	操作系统
5、6	网络基础	实验	实验	实验	



# 表格与列表标记

## 2、列表标记

三种类型的列表：

- 👉 **无序列表(Unordered list):** `<ul>列表项</ul>`
  - 👉 **有序列表(Ordered list):** `<ol>列表项</ol>`
  - 👉 **定义列表(Description list):** `<dl>列表项</dl>`
- 
- 👉 列表中每一项以标记`<li>`开始，以标记`</li>`结束。列表可嵌套使用，一个列表中的列表项又可以是一个列表。



# 表格与列表标记

## 无序列表(Unordered list)

无序列表<ul>具有**type**属性， type取值：

**type = disk**          加重符号是实心圆点(缺省)

**type = circle**        加重符号是空心圆点

**type = square**        加重符号是实心方块



# 表格与列表标记的应用

<HTML>

<HEAD><TITLE>无序列表标记的使用</TITLE></HEAD>

<BODY><H4>无序列表1: 属性TYPE为缺省值</H4>

<UL>

<LI><H5>第一项。</H5></LI>

<LI><H5>第二项。</H5></LI>

</UL>

<H4>无序列表2: 属性TYPE=circle</H4>

<UL TYPE=circle>

<LI><H5>第一项。</H5></LI>

<LI><H5>第二项。</H5></LI>

</UL>

<H4>无序列表3: 属性TYPE=square</H4>

<UL TYPE=square>

<LI><H5>第一项。</H5></LI>

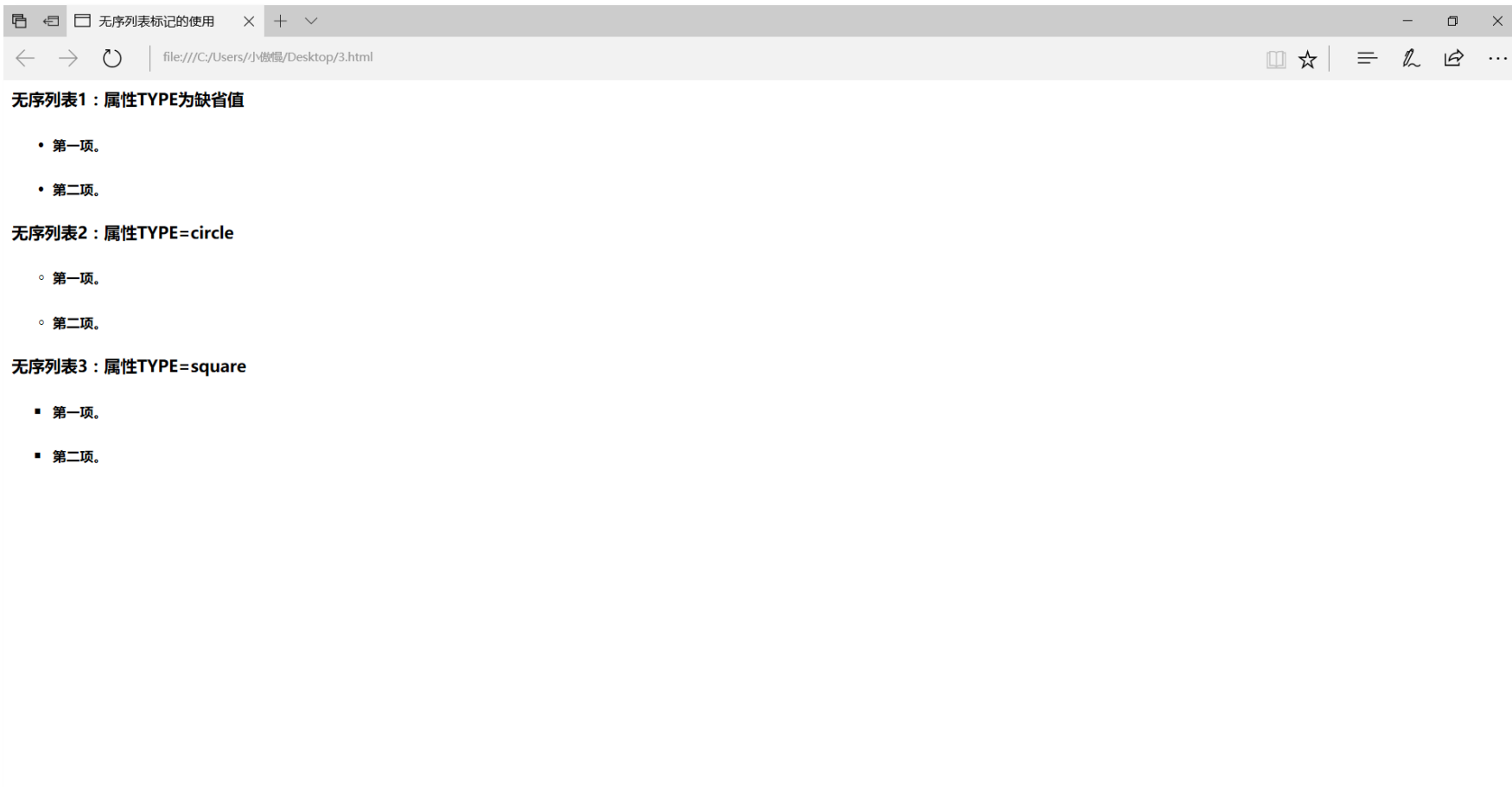
<LI><H5>第二项。</H5></LI>

</UL>

</BODY></HTML>



# 表格与列表标记





# 表格与列表标记

## 有序列表(Ordered list)

👉 有序列表标记<ol>，列表中序列号由浏览器自动给出。

**type属性：**

**type = 1**，缺省值，用数字1, 2, 3等标识各项

**type = A**，用大写字母A, B, C等标识各项

**type = a**，用小写字母a, b, c等标识各项

**type = I**，用大写罗马字母标识各项

**type = i**，用小写罗马字母标识各项

**start属性：** 指定列表开头数字

如 **type = A, start = 3**，列表的第一项 从C开始

👉 <li>标记可以具有属性value，将列表编号指定为特定值。





# 表格与列表标记的应用

<HTML>

<HEAD><TITLE>有序列表标记的使用</TITLE></HEAD>

<BODY><H3>有序列表的属性</H3>

<OL TYPE=I>

<LI>

<H4>有序列表标记具有属性START。用START属性指定列表编号的开头值。

</H4>

<LI><H4>有序列表标记具有属性TYPE</H4>

<OL TYPE=1 START=6>

<LI><H5>TYPE=1, 缺省值, 用数字1, 2, 3等标识各项。</H5></LI>

<LI><H5>TYPE=A, 用大写字母A, B, C等标识各项。</H5></LI>

<LI><H5>TYPE=a, 用小写字母a, b, c等标识各项。</H5></LI>

<LI><H5>TYPE=I, 用大写罗马字母标识各项。</H5></LI>

<LI VALUE=3><H5>TYPE=i, 用小写罗马字母标识各项。</H5></LI>

</OL>

</OL>

</BODY></HTML>



# 表格与列表标记

无序列表标记的使用

有序列表标记的使用

file:///C:/Users/小傲慢/Desktop/4.html

有序列表的属性

I. 有序列表标记具有属性START。用START属性指定列表编号的开头值。

II. 有序列表标记具有属性TYPE

6. TYPE = 1, 缺省值, 用数字1, 2, 3等标识各项。

7. TYPE = A, 用大写字母A, B, C等标识各项。

8. TYPE = a, 用小写字母a, b, c等标识各项。

9. TYPE = I, 用大写罗马字母标识各项。

3. TYPE = i, 用小写罗马字母标识各项。



# 表格与列表标记

## 定义列表(Description list)

定义列表标记<dl>

### 组成

术语：<dt>标记描述

术语的定义：<dd>标记描述

### 格式

<dl>

<dt>术语<dd>术语的定义1

<dd>术语的定义2

<dt>术语<dd>术语的定义

</dl>



# 表格与列表标记的应用

**<HTML><HEAD>**

**<TITLE>**定义标记的使用**</TITLE></HEAD>**

**<BODY><H3>**定义列表的使用**</H3>**

**<DL>**

**<DT><B>**列表**</B><DD>**数据项的有序集合。

**<DT><B>**标号**</B><DD>**程序中指令的标识符。

**<DD>**磁盘文件的标识说明记录。

**<DD>**数据集中或附加在数据集上的字符，  
它包括数据集的信息。

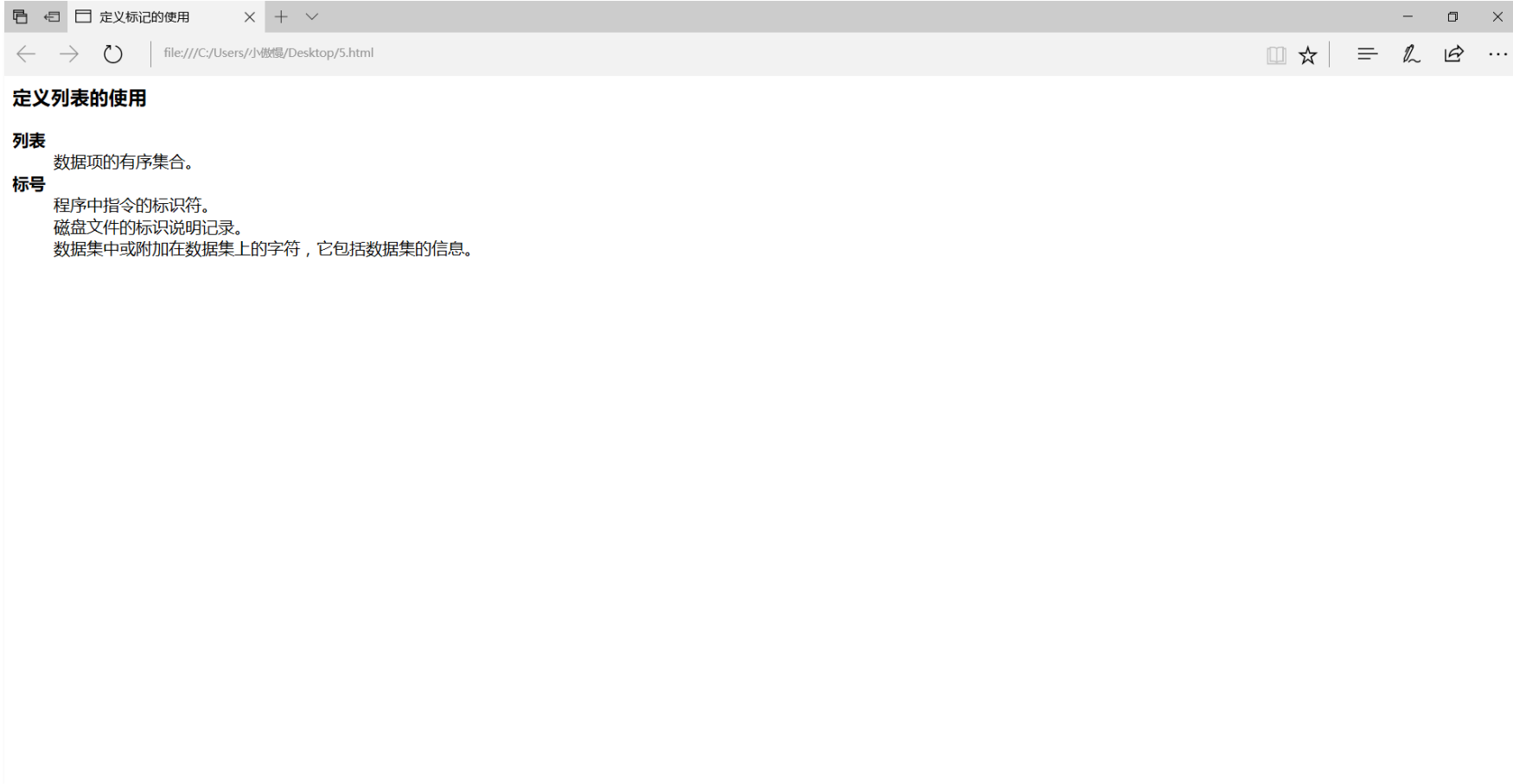
**</DL>**

**</BODY>**

**</HTML>**



# 表格与列表标记





# 超链接标记

超链接标记是超文本的基本结构，把不连续的文字或文件连接起来。网页上的超链接一般指同一网页和不同网页之间的链接。

- 👉 **`<a href="URL">链接文本或图像</a>`**
- 👉 **href 属性：**指明所要链接资源的URL，
- 👉 **`<a>`和`</a>`之间的内容是链接点，** 点击链接点网页转跳至**href**指明的资源处。



# 超链接标记的应用

---

```
<html><head>  
<title>超链接标记应用</title>  
</head>  
<body><center>  
<a href="15.html">下一页</a><p>  
<a href="18.html">  
</a><p>  
超链接标记的功能</center>  
</body>  
</html>
```



# 超链接标记

## 锚——同一页面间的链接

实现同一网页内部的链接，指向同一网页的特定位置，相当于提供了一个屏幕滚动功能。格式为：

`<a href="#锚名">链接文本</a>`

`<a name="锚名">“锚” 文本</a>`

锚名是网页中特定位置的名称。





# 超链接标记

## 锚的应用

```
<html><head>
```

```
  <title>同一页面间的链接</title>
```

```
</head>
```

```
<body>
```

```
<h1>Web技术应用基础</h1>
```

```
<a href="#第1章">第1章 Web技术概述</a><p>
```

```
<a href="#第2章">第2章 Web应用环境构建技术</a><p>
```

```
<a href="#第3章">第3章 基于Web方式的信息系统开发案例 </a><p>
```

```
<a name="第4章标题" href="#第4章" >第4章 HTML</a><p>
```

```
<a href="#第5章">第5章 CSS</a><p>
```

```
<a href="#第6章">第6章 XML<p>
```

```
<a href="#第7章">第7章 JavaScript</a><p>
```

```
<a href="#第8章">第8章 Web数据库编程技术</a><p>
```



# 超链接标记

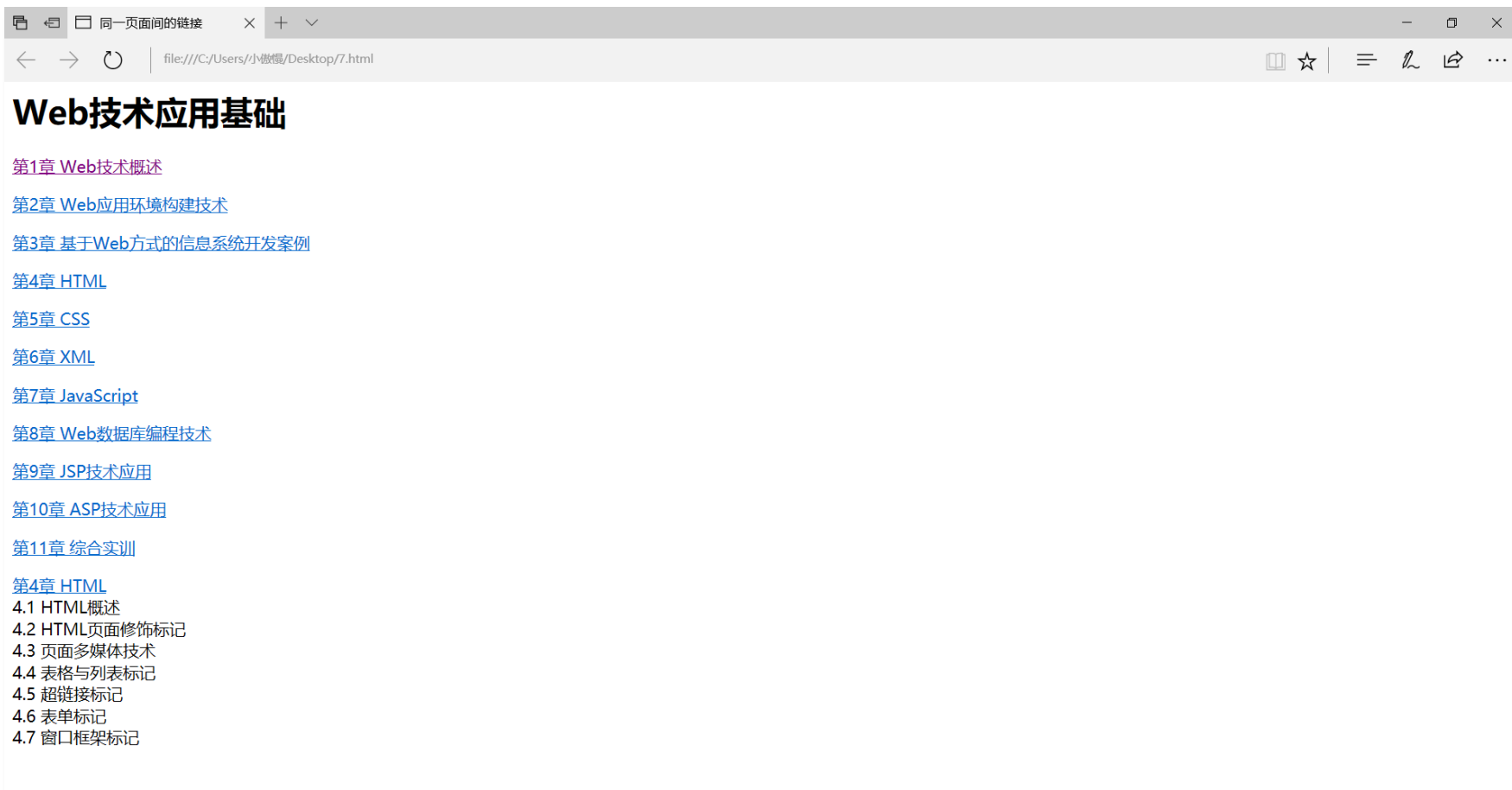
## 锚的应用

```
<a href="#第9章">第9章 JSP技术应用</a><p>  
<a href="#第10章">第10章 ASP技术应用</a><p>  
<a href="#第11章">第11章 综合实训</a><p>  
<a name="第4章" href="#第4章标题">第4章 HTML</a><br>  
4.1 HTML概述<br>  
4.2 HTML页面修饰标记<br>  
4.3 页面多媒体技术<br>  
4.4 表格与列表标记<br>  
4.5 超链接标记<br>  
4.6 表单标记<br>  
4.7 窗口框架标记<br>  
</body></html>
```



# 超链接标记

## 锚的应用





# 超链接标记

## 链接电子信箱

语法规则：

`<a href="mailto:电子信箱地址">电子信箱地</a>`

如：

`<a href="mailto:libadmin@mail.nnb.edu.cn">`

`libadmin@mail.nnb.edu.cn</a>`

建立了系统管理员电子信箱超链接，点击信箱地址，向系统管理员发信。



# 表单标记

## 1、表单的作用




- ✎ 提供图形用户界面和用户输入数据的元素：基本元素有按钮、文本框、单选钮、复选框等，是实现交互功能的主要接口。
- ✎ 用户通过表单向服务器提交信息，表单接收用户信息，并把信息提交给服务器，由服务器端的应用程序处理用户信息，并把处理结果返给用户并向用户显示。



# 表单标记

## 2、表单标记

`<form method="post/get" action="URL" enctype="application/x-www-form-urlencoded"> </form>`，属性有：

-  **action:** 完成表单信息处理任务服务器程序的完整URL，如 `http://www.buu.edu.cn/ReaderNote.asp`，可以是CGI、ASP、PHP、JSP等程序。
-  **method:** 表单中输入数据的传输方法，缺省值get。method = get，将表单中输入数据打包后加到在action中指定的URL后面；而post将表单中的数据和调用程序分开发送，一般来讲使用post比较安全。
-  **enctype:** 指定表单中输入数据的编码方法。



# 表单标记

## 3、<input>输入标记



定义输入控件，类型由type属性确定

button: 定义可点击按钮

checkbox: 定义复选框

file: 定义输入字段和“浏览”按钮，供文件上传

hidden: 定义隐藏的输入字段

image: 定义图像形式的提交按钮

password: 定义密码字段，该字段中的字符被掩码

radio: 定义单选按钮

reset: 定义重置按钮，清除表单中的所有数据

submit: 定义提交按钮，把表单数据发送到服务器

text: 定义单行的输入字段，默认宽度为 20 个字符



# 表单标记

## 3、<input>输入标记



属性有：

**type:** 控件类型，缺省值是text

**name:** 控件标识

**value:** 控件输入域的初始值

**maxlength:** 控件输入域允许输入最多字符数

**size:** 控件输入域大小

**checked:** 复选框和单选按钮初始状态

**URL:** 图像按钮使用图像位置(URL)

**align:** 图像的对齐方式





# 表单标记

## 4、列表框<select>标记



定义下拉式列表框和滚动式列表框，属性：

**name:** 列表框名字

**size:** 列表框大小，用户一次可见列表项数目

**multiple:** 允许用户进行多项选择



### <option>标记

定义<select>标记所定义的列表框中的各个选项，属性：

**selected:** 表示该项预先选定

**value:** 指定控件初始值



# 表单标记的应用

---

```
<html>
```

```
<head><title>表单标记的应用</title></head>
```

```
<body>
```

```
<form method="post" action="travel.asp">
```

请输入姓名:

```
<input type=text name="name1" size=12 maxlength=6><p>
```

性别:

```
<select name="性别">
```

```
  <option> 男
```

```
  <option> 女
```

```
</select><p>
```



# 表单标记的应用

请选择旅游城市，可做多项选择

`<input type=checkbox name="复选框1" checked>` 北京

`<input type=checkbox name="复选框2" >` 上海

`<input type=checkbox name="复选框3" >` 西安

`<input type=checkbox name="复选框4" >` 昆明`<p>`

请选择付款方式

`<input type=radio name="单选按钮1" >` 信用卡

`<input type=radio name="单选按钮1" checked>` 现金`<p>`

`<input type=reset name="复位按钮" value="复位">`

`<input type=submit name="提交按钮" value="确定">`

`</form>`

`</body>`

`</html>`



# 表单标记的应用

表单标记的应用

file:///C:/Users/小傲慢/Desktop/1.html

请输入姓名:

性别:

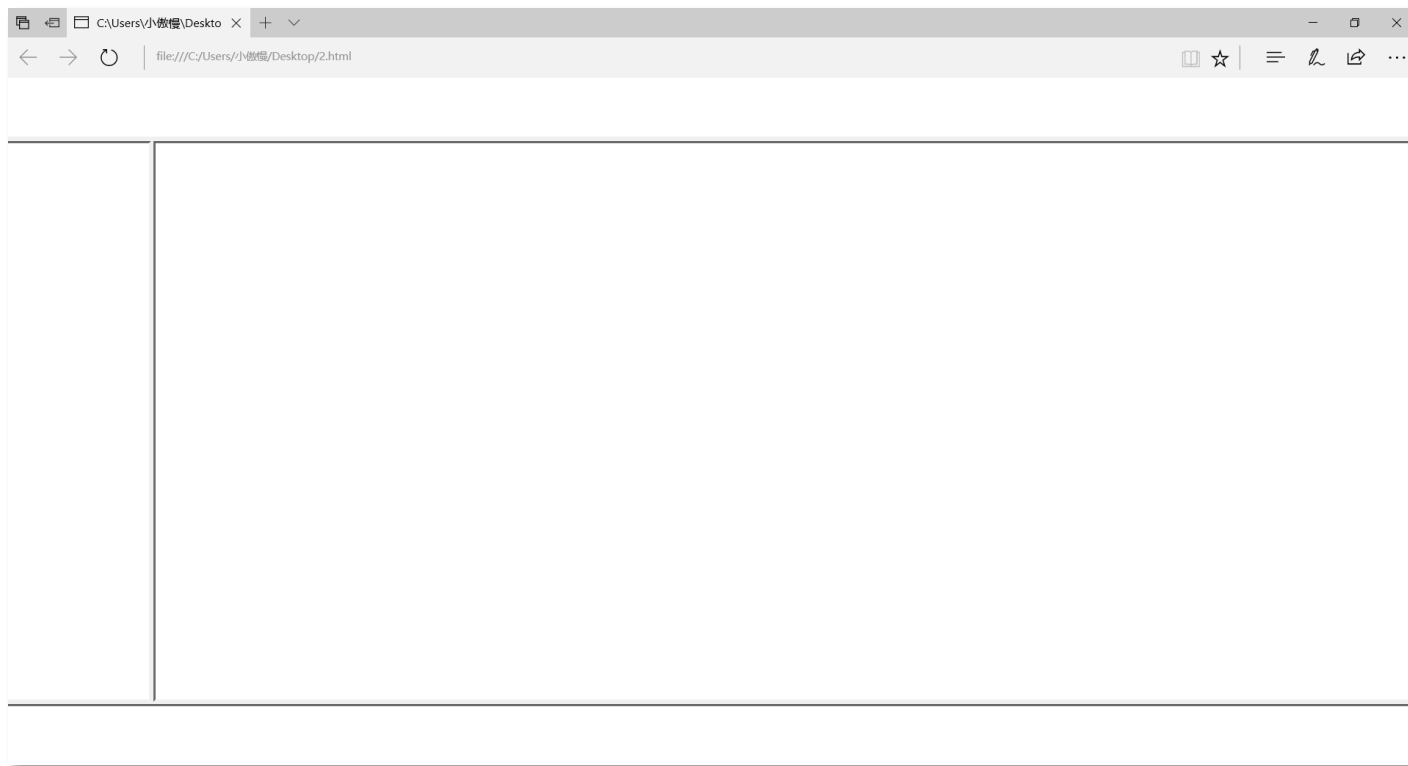
请选择旅游城市, 可做多项选择 ☒ 北京 ☐ 上海 ☐ 西安 ☐ 昆明

请选择付款方式 ☐ 信用卡 ☒ 现金



# 窗口框架标记

**帧（Frame）**：可用来将浏览器窗口划分为多个区域（子窗口），每个子窗口中装载一个HTML文件，即每个HTML文件占据一个帧，而多个帧可以同时显示在同一个浏览器窗口中，这样的Web页面称为**框架网页**。





# 窗口框架标记

## <frameset>标记

- ✎ 用来定义主文档中有几个帧并且各个帧是如何排列的；
- ✎ 具有rows和cols属性，使用<frameset>标记时这两个属性至少必须选择一个，否则浏览器只显示第一个定义的帧；
- ✎ 放在帧的主文档的<body></body>标记对的外边；
- ✎ 可以嵌在其它帧文档中，并可嵌套使用；
- ✎ rows用来规定主文档中各个帧的行定位；



# 窗口框架标记

## <frameset>标记

- 👉 cols用来规定主文档中各个帧的列定位;
- 👉 这两个属性的取值可以是百分数、绝对像素值或星号(“\*”), 其中星号代表那些未被说明的空间;
- 👉 如果同一个属性中出现多个星号则将剩下的未被说明的空间平均分配;
- 👉 所有的帧按照rows和cols的值从左到右, 从上到下排列。



# 窗口框架标记

## <frame>标记

- ✎ 放在<frameset>...</frameset>之间，定义具体的帧；
- ✎ 具有src和name属性，这两个属性都是必须赋值的；
- ✎ src是此帧的源HTML文件名(包括网络路径，即相对路径或网址)，浏览器将会在此帧中显示src指定的HTML文件；
- ✎ name是此帧的名字，这个名字是用来供超文本链接标志<a href="" target="">中的target属性用来指定链接的HTML文件将显示在哪一个帧中；
- ✎ <a href="newone.htm" target="main">链接的文本</a>





# 窗口框架标记

## <noframes>标记

✎ 有的浏览器不支持框架网页，此时需要使用  
<noframes></noframes>标记对，用来在那些不支  
持帧的浏览器中书写传统的<body>...</body>部分。



# 窗口框架标记的应用

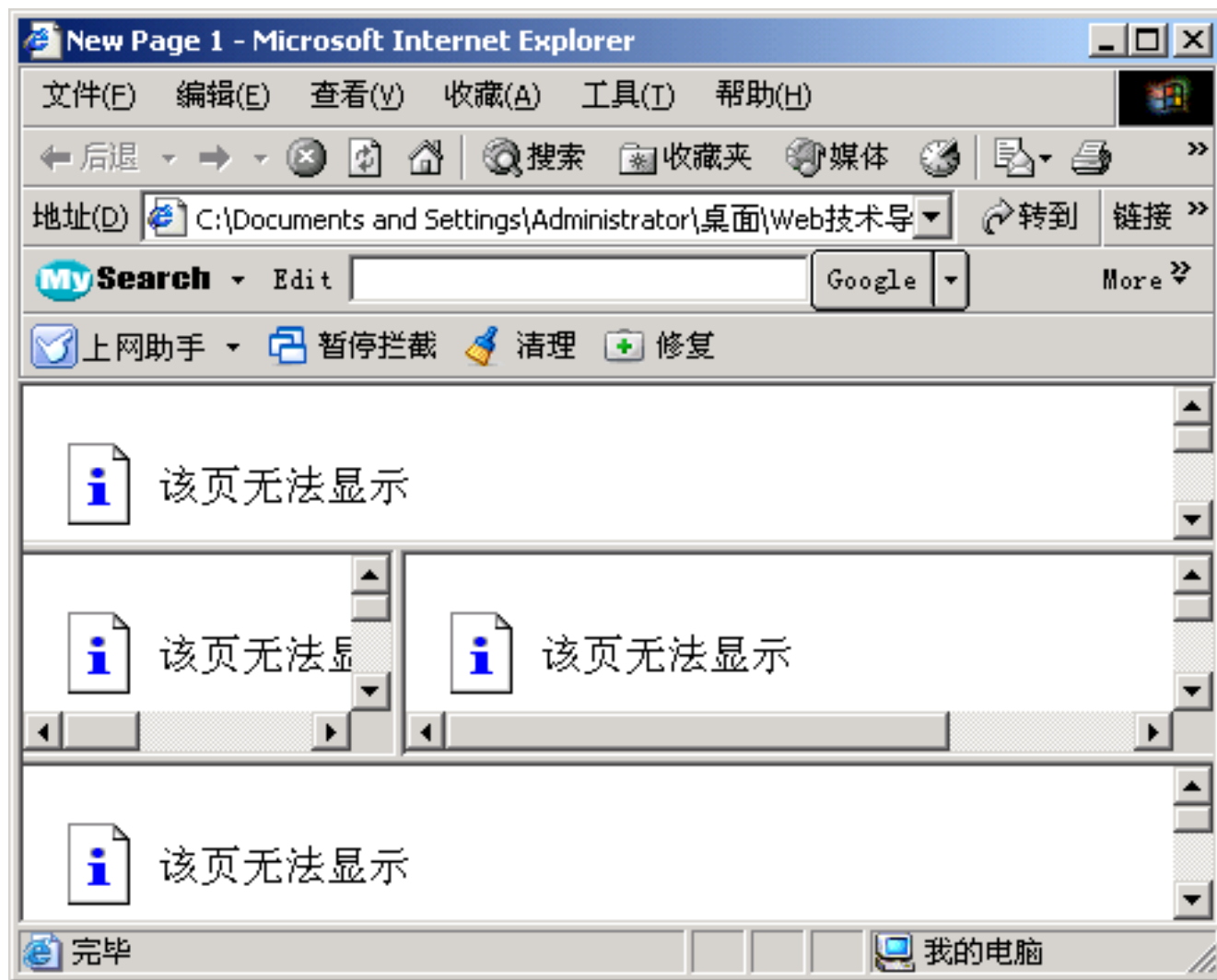
## 帧 (frame) - 举例

```
<html>
<head>
</head>
<frameset rows="64,*,64">
  <frame name="top" src="new_page_2.htm">
  <frameset cols="150,*">
    <frame name="contents" src="new_page_3.htm">
    <frame name="main" src="new_page_4.htm">
  </frameset>
  <frame name="bottom" src="new_page_5.htm">
</frameset>
<noframes>
<body>
<p>此网页使用了框架，但您的浏览器不支持框架。</p>
</body>
</noframes>
</frameset>
</html>
```



# 窗口框架标记的应用

## 帧的显示效果





# 窗口框架标记

## 文件头及相关标记

**<head>**与**</head>**标记对之间的部分称为文件头



**<title></title>**标记



**<meta>**标记



**<base>**标记



**<link>**标记



**<isindex>**标记



注释标记



# 窗口框架标记

## <meta>标记

- ✎ 它是HTML文档文件头<head>...</head>标记内的一个辅助性标记，往往不引起用户的注意，但是它对于网页是否能够被搜索引擎检索、提高网页在搜索列表的排序起着关键的作用。
- ✎ <meta>标记为单标记，没有尾标记。共有两个属性：  
**http-equiv**属性和**name**属性。



# 窗口框架标记

## <meta>标记—name属性

👉 name属性主要用于描述网页，对应的属性值为content

<meta name= “参数” content= “具体的参数值” >

👉 其中name属性主要有以下几种参数值：

**keywords:** 告诉搜索引擎该网页的关键字是什么

**description:** 告诉搜索引擎网站的主要内容

**author:** 标注网页的作者

**robots:** 告诉搜索机器人需要索引的页面有哪些



# 窗口框架标记

## <meta>标记—name属性（示例）



```
<meta name="keywords"
content="science,education,culture,entertainment ">
```



```
<meta name="description"
content="This page is about science, education etc.">
```



```
<meta name="author" content="brion@mail.abc.com">
```



```
<meta name="robots" content="none">
```



# 窗口框架标记

## <meta>标记—name属性



当name属性值为robots时，content取值：

**all:**文件将被检索，且页面上的链接可被查询，默认值

**none:**文件将不被检索，且页面上的链接不可被查询

**index:**文件将被检索

**follow:**页面上的链接可被查询

**noindex:**文件将不被检索，但页面上的链接可被查询

**nofollow:**页面上的链接不可被查询





# 窗口框架标记

## <meta>标记—http-equiv属性



相当于HTTP的文件头作用，可以向浏览器传回一些有用的信息，以帮助正确显示网页内容

**<meta http-equiv="参数" content="参数变量值">**

**content-type:** 设定页面使用的字符集

**expires:** 设定网页的到期时间，一旦过期，必须重新下载

**pragma:** 禁止浏览器从本地计算机的缓存中访问页面内容

**refresh:** 自动刷新并指向新页面

**set-cookie:** 如果网页过期，那么存盘的cookie将被删除

**window-target:** 设定显示窗口



# 窗口框架标记

## <meta>标记—http-equiv属性（示例）



`<meta http-equiv="content-type" content="text/html; charset=gb2312">`



`<meta http-equiv="expires" content="Fri, 12 Jan 2017 18:18:18 GMT">`



`<meta http-equiv="pragma" content="no-cache">`



`<meta http-equiv="refresh" content="60; url=new.htm">`



`<meta http-equiv="set-cookie" content="cookievalue=xxx;expires=Friday, 12-Jan-2004 20:30:00 GMT; path=/">`



`<meta http-equiv="window-target" content="_top">`



# 窗口框架标记

## <base>标记



**href 属性：** 指定文档的基础URL地址



**target 属性：** 同框架一起使用，它定义了当文档中的链接被点击后，在哪一个框架中展开页面。

**\_blank：** 表明在新窗口中打开链接指向的页面

**\_self：** 在当前文档的框架中打开页面

**\_parent：** 在当前文档的父窗口中打开页面

**\_top：** 在链接所在的完整窗口中展开页面



# 窗口框架标记

## <link>标记



定义了文档之间的链接关系

**<link href="'url" rel="relationship">**



**href属性:**指向链接资源所在的URL






**rel属性:**定义了文档和所链接资源的链接关系，可能的值包括：alternate, stylesheet, start, next, prev, contents, index, glossary, copyright, chapter, section, subsection, appendix, help, bookmark等。



# 窗口框架标记

## <isindex>标记

-  **< isindex prompt="输入搜索内容">**
-  在Web页面的开始则出现“输入搜索内容”，后面有一个文本框，用于输入搜索的内容。
-  <isindex>标记并不是只能用于html文档头部，它还可以用于表单（form）内。



# 窗口框架标记

## 网页的基本元素

**文本：**在网页上的文字信息。

**表格：**把信息分类显示，使大量信息一目了然。

**表单：**提供用户与页面交互功能。

**框架：**把浏览器窗口分成不同帧，显示不同网页

**图像：**网页中图像增加页面表现力，过多图像影响页面浏览速度。

**超链接：**超链接可以是图像或文字。

**动态元素：**使网页生动富有动感，常用的有：**GIF**动画、滚动字幕、网站计数、动态视频等。



# 层叠样式表CSS

- 👉 **背景：**HTML显示特性通过标记属性设置，一旦设置不能随意改变版面和编排文字，也不能由程序控制。
- 👉 W3C 协会公布了一套HTML的扩展标准 **CSS** (Cascading Style Sheets)，扩展了HTML在排版和文字样式上的功能,可以通过**脚本程序控制**，使页面表现方式更加灵活，具有动态特性。



# 层叠样式表CSS

## 1、CSS的基本概念

- 👉 HTML网页要展示的样式在**标记内设定**，样式分散在各个标记中，在更改样式时，需要逐个修改各个标记中的属性。

`<h2><font color=red>你好!</h2>`

- 👉 **CSS:** 把网页展现的样式从网页中独立出来集中管理，如果需要改变网页样式，只需要改变样式设定部分，HTML文件本身无须更改。

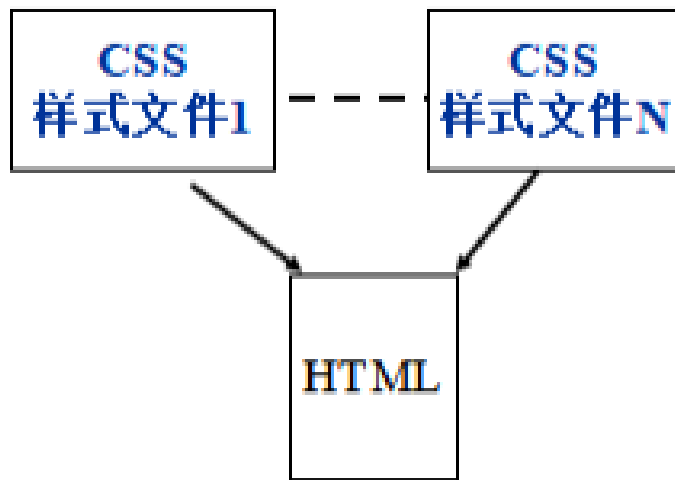
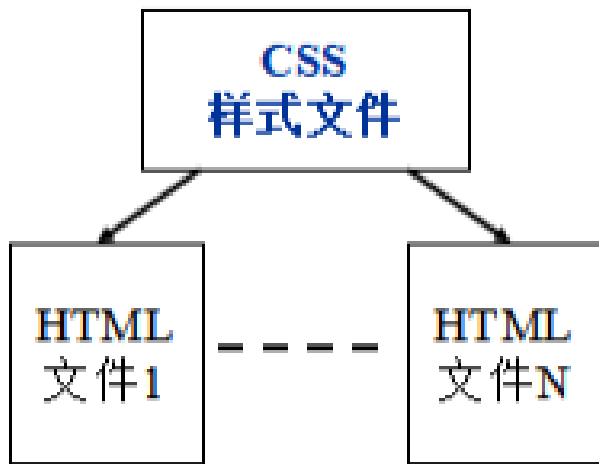




# 层叠样式表CSS

## 2、样式表的使用方式

- 👉 页面样式**设定**与页面**内容**分离,把CSS样式信息存成独立文件，使多个网页文件共享样式文件。
- 👉 把样式分类，分存于不同的文件，如分为编排样式文件、字体样式文件、颜色样式文件等，把多个样式文件套用在一个网页文件上。







# 层叠样式表CSS

## 3、定义样式格式

选择器 { 规则 }

如: `h1{ color:blue;}`

 **选择器(Selector):** 样式要套用的对象，一般是HTML标记，如h1。在文件<h1>...</h1>标记之间的内容将全部继承h1的全部规则。

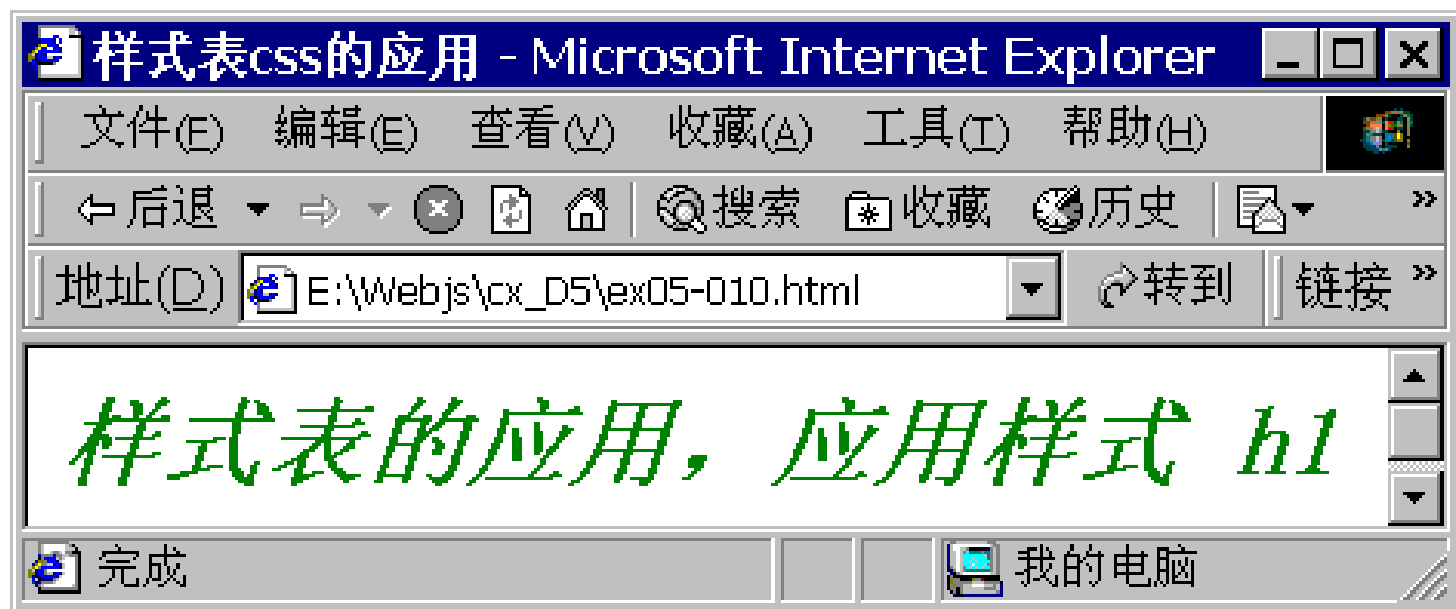
 **规则(Rule):** 是样式设定的内容，是用{}括起来的部分，如{ color:blue;}



# 层叠样式表CSS

## 4、样式表的应用

从CSS 样式代码插入的形式来看可以分为：**内联式样式表**、**嵌入式样式表**、**外部(链接)样式表**。





# 层叠样式表CSS

## 4、样式表的应用

### I 内联式样式表

👉 内联式样式表就是把css代码直接写在现有的HTML标记中

`<h1 style="color:green;text-align:center;font-style:italic;font-size:x-large">样式表的应用，应用样式 h1</h1>`

👉 缺点：

内容和显示混杂；

内联样式只针对当前标记，对页面上其它任何同名标记不起作用，如果多个标记使用同样的样式显示时需要为每个标记添加style属性。



# 层叠样式表CSS

## 4、样式表的应用

### II 嵌入式样式单

把样式定义放在文档头部，<style></style>标记之间定义，整个页面都以该样式显示。

```
<html><head><title>样式表css的应用</title>
```

```
<style type="text/css"> /*定义样式*/
```

```
  h1{  color:green;
        text-align:center;
        font-style:italic;
        font-family:楷体;
        font-size:x-large;
      }
```

```
</style></head>
```

```
<body><h1>样式表的应用，应用样式 h1</h1>
```

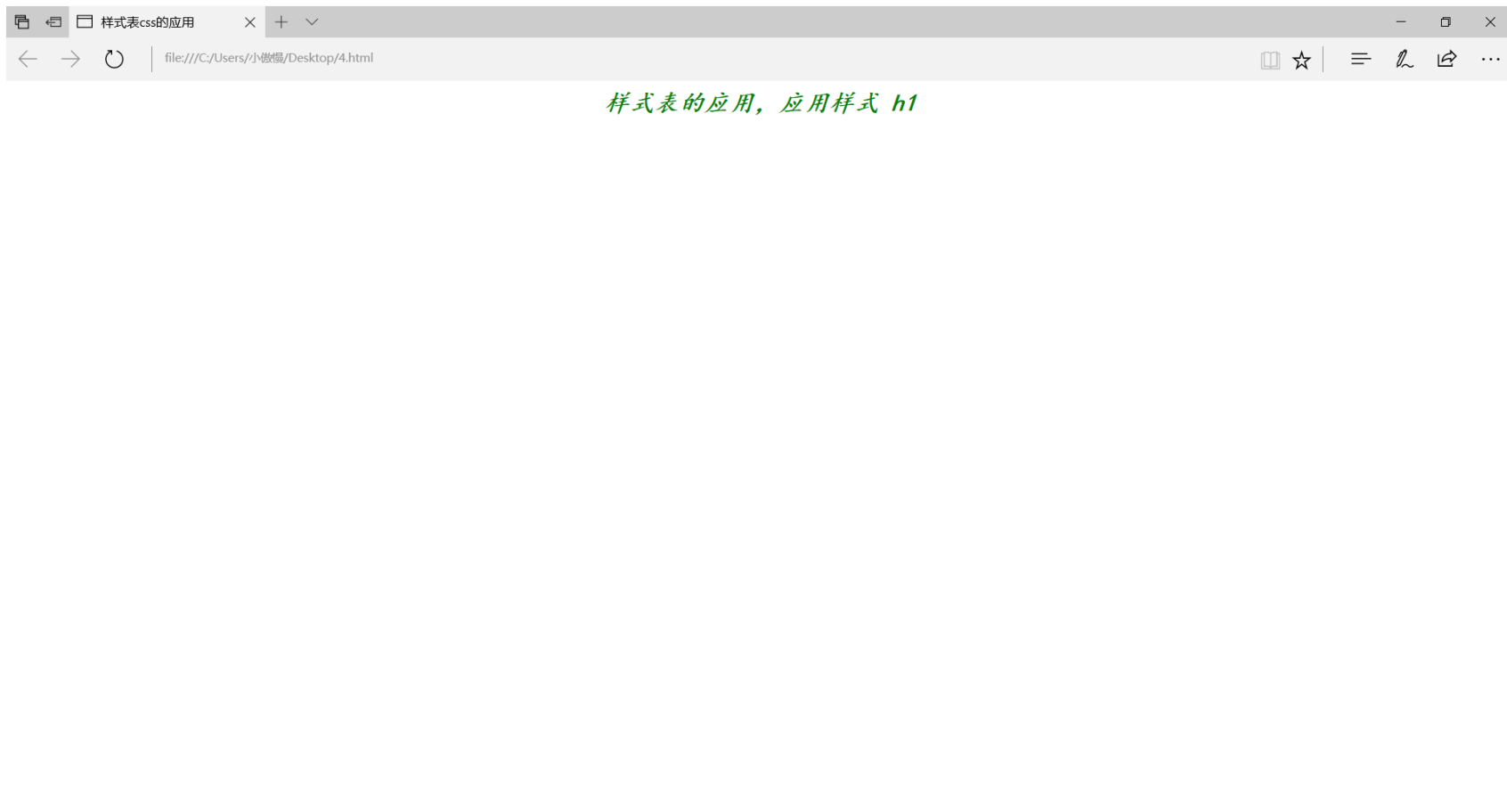
```
</body></html>
```



# 层叠样式表CSS

## 4、样式表的应用

### II 嵌入式样式单的显示效果





# 层叠样式表CSS

## 4、样式表的应用

### III 外部（链接）样式单

当多个网页具有相同样式时，可以使用样式文件（后缀是.css）把设定的样式集中起来，使多个网页共享该样式文件。也可以将样式分类，使一个网页套用多个样式文件。



# 层叠样式表CSS

## 4、样式表的应用

### III 外部（链接）样式单

两个样式文件：

ex05-010\_1.css 保存文字的颜色

ex05-010\_2.css 保存文字的其它样式

ex05-010\_1.css文件代码清单：

```
h1 { color:green;}
```

ex05-010\_2.css文件代码清单下：

```
h1 { text-align:center;  
      font-style:italic;  
      font-family:楷体;  
      font-size:x-large;  
}
```





# 层叠样式表CSS

## 4、样式表的应用

### III 外部（链接）样式单

在HTML文档的文件头中加入语句，建立与外部样式文件的链接。

`<link rel=stylesheet type="text/css" href="样式文件的URL">`

链接外部样式文件的HTML文件的页面代码

`<html>`

`<head>`

`<title>样式表css应用</title>`

`<link rel=stylesheet type="text/css" href="ex05-010_1.css">`

`<link rel=stylesheet type="text/css" href="ex05-010_2.css">`

`</head>`

`<body>`

`<h1>样式表的应用，应用样式 h1</h1>`

`</body>`

`</html>`



## 1、DHTML技术

**DHTML**即动态的HTML语言(Dynamic HTML)。DHTML并不是一门新的语言，它是以下技术、标准或规范的一种集成：

**HTML 4.0**

**CSS** (Cascading Style Sheets, 层叠样式单)

**CSSL** (Client-Side Scripting Language, 客户端脚本语言)

**HTML DOM** (Document Object Model, HTML文档对象模型)



## 2、DHTML基本结构

DHTML基于HTML，通过CSS技术扩展HTML在样式编排上的不足，使用脚本程序调用或控制浏览器对象和网页对象，使页面具有动态效果和交互功能。





## 3、DHTML特点

- 👉 将页面内容对象化，把网页上的图像和文字看作对象；
- 👉 通常使用VBScript或JavaScript脚本程序把这些对象组装到一起，实现其动态和交互性；
- 👉 网页无须重新下载，即可动态更新网页内容和显示式样；
- 👉 所有的功能都是在用户的计算机上实现并完成的，因此提高了响应速度；
- 👉 突破了传统的静态Web页面的局限。



## 4、DHTML应用

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
function bgChange(bg)
```

```
{
```

```
document.body.style.background=bg
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<b>请把鼠标移动到正方形上，背景颜色会发生改变！</b>
```

```
<table width="300" height="100">
```

```
<tr>
```



## 4、DHTML应用

```
<td onmouseover="bgChange('red')"  
    onmouseout="bgChange('transparent')"  
    bgcolor="red">
```

```
</td>
```

```
<td onmouseover="bgChange('blue')"  
    onmouseout="bgChange('transparent')"  
    bgcolor="blue">
```

```
</td>
```

```
<td onmouseover="bgChange('green')"  
    onmouseout="bgChange('transparent')"  
    bgcolor="green">
```

```
</td>
```

```
</tr>
```

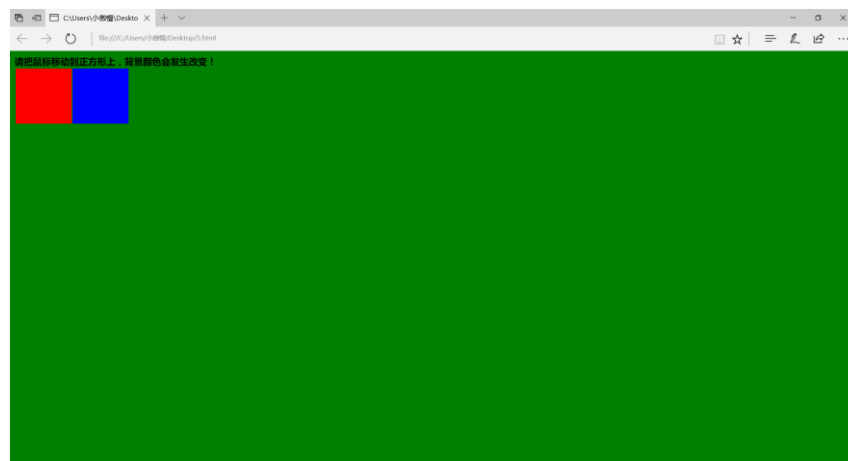
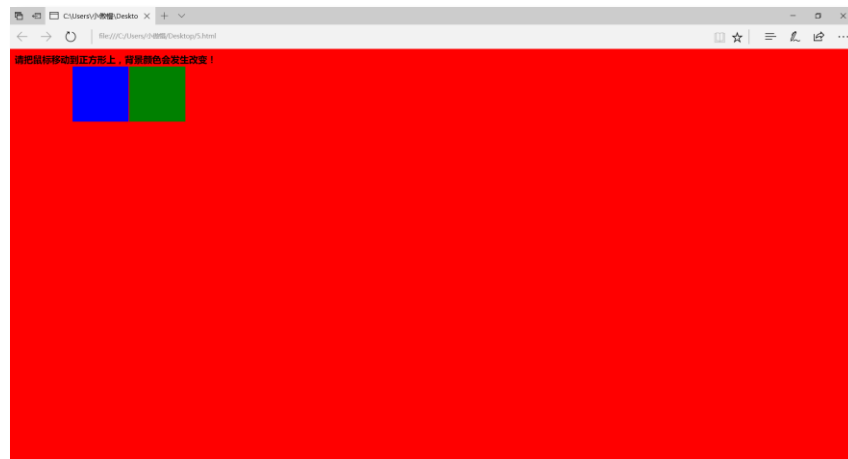
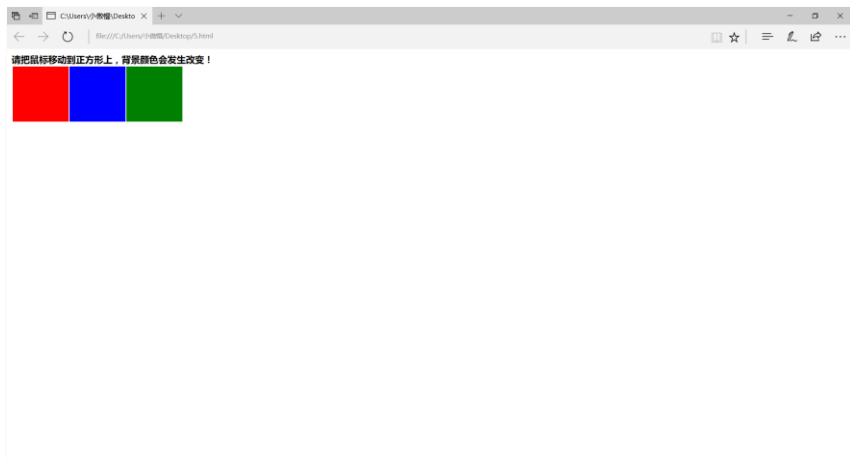
```
</table>
```

```
</body>
```

```
</html>
```



## 4、DHTML应用显示效果





# XML

---

## 1、XML定义

- ✎ **XML(eXtensible Markup Language):** 可扩展的标记语言，可以定义其他语言的语言。
- ✎ **包括一组相关技术:** XSL(可扩展样式语言)、XML链接语言、XML名称空间、XML模式(Schema)





## 2、XML特点

- ✎ XML把显示**内容**与显示**格式**分开。显示内容在XML文件中，显示方式在样式文件CSS或XSL中。使XML文件的制作者集中精力于**数据本身**；
- ✎ 不同的样式文件，使相同的数据以**不同的方式显示**，以利于数据的**重用**；
- ✎ 减少数据的传送量。



## HTML:

数据显示，信息如何在浏览器中展示

HTML标记是给定的，用户不能增加新标记

## XML:

组织和描述数据，XML文档基本上不涉及数据显示方式。

XML的标记是可扩展的，用户可以根据应用需要定义新的标记。



## 3、XML文件的结构

### I 文件头

文件头是XML说明部分，它描述了XML文档、版本号、编码信息和其他一些信息。

**<?xml version="1.0" encoding="GB2312"?>**

**<?.....?>**: 处理指令，以“<?”开始，“?>”结束

**xml**: 在“<?”后面的xml说明该文件是XML文件

**version="1.0"**: 遵循XML1.0规范

**encoding="GB2312"**: 应用中文GB2312字符编码



## 3、XML文件的结构

### II 文件体

文件体用来存放XML文件中被应用程序使用的信息

### III 注释

XML中注释在<!--.....-->之间

## 4、XML的功能

异种数据之间的交互

数据的多样化显示

分布式计算

将数据内容与内容的表现方式分割



## 5、XML文档的书写规则

### I 区分大小写

`<?xml.....?>` ✓

`<?xmL.....?>` ✗

`<?XML.....?>` ✗

关键字如“xml”必须严格按照要求来书写。用户自定义的标记可以大小写混用，但是必须配对。

`<Person>`和`</Person>` ✓

`<Person>`和`</person>` ✗



# XML

## 5、XML文档的书写规则

### II 文件第一行必须是xml 声明

`<?xml version="1.0" encoding="GB2312"?>`

`<?xml-stylesheet type="text/css" href="ex08_001.css"?>`

`<personlist>`

`<person>`



`<! -- person data -- >`

`<?xml version="1.0" encoding="GB2312"?>`

`<?xml-stylesheet type="text/css" href="ex08_001.css"?>`

`<personlist>`

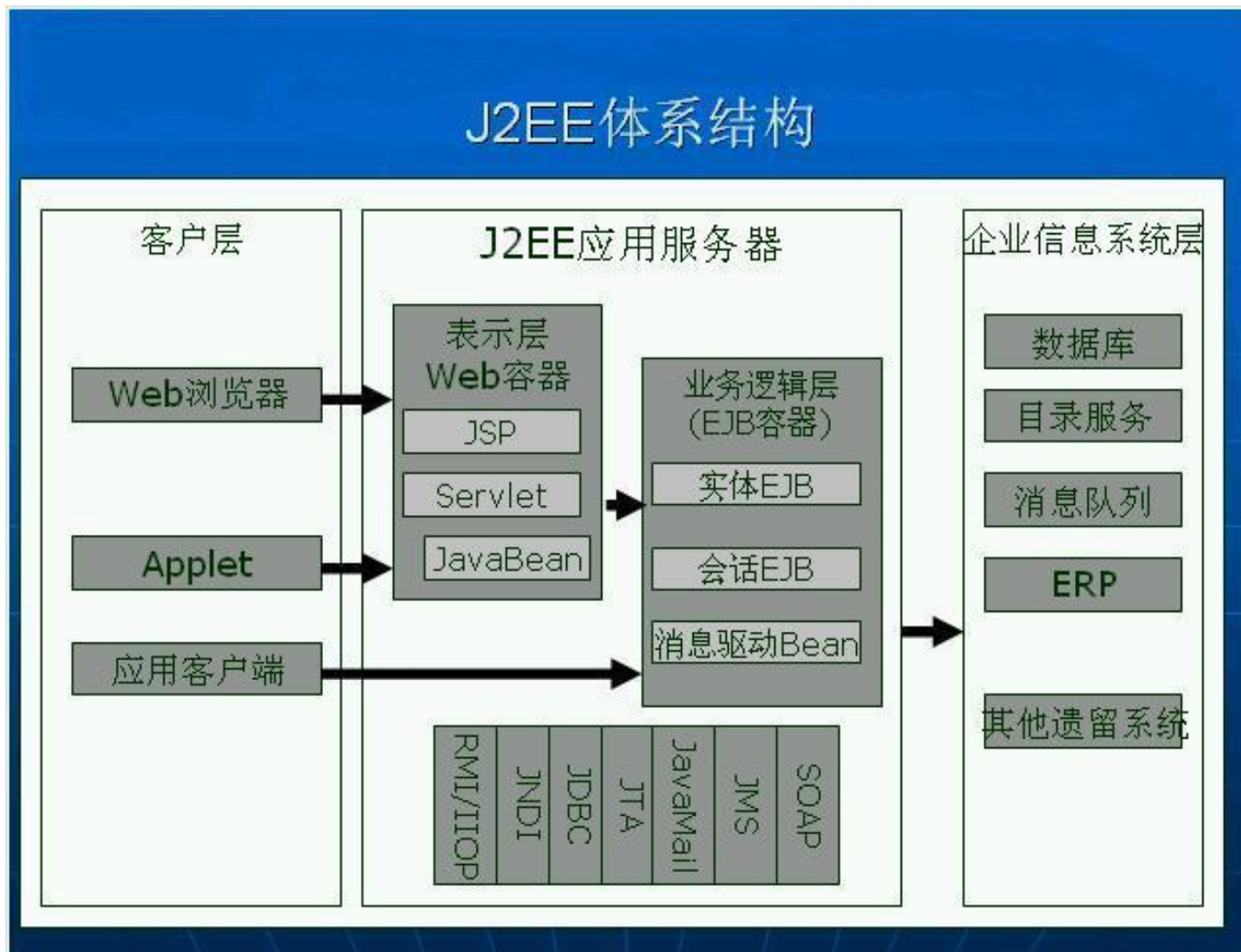
`<person>`





# J2EE

## J2EE体系结构





## 1、J2EE组件



**J2EE组件**是一个由**Java**语言编写、封装了功能的软件单元，能够与相关的一些类和文件一起组成**J2EE**应用程序，可简化且规范应用系统的开发与部署，进而提高可移植性、安全与再用价值。



**J2EE组件**和“标准的”**Java**类的不同点在于：它被装配在一个**J2EE**应用程序中，具有固定的格式并遵守**J2EE**规范，由**J2EE**服务器对其进行管理。





## 1、J2EE组件

J2EE规范是这样定义J2EE组件的：

- 👉 客户端应用程序和applet是运行在客户端的组件。
- 👉 Java Servlet和Java Server Pages (JSP)是运行在服务器端的Web组件。
- 👉 Enterprise JavaBean(EJB)组件是运行在服务器端的业务组件。



## 1、J2EE组件

### J2EE客户端组件

- 👉 Applets(客户端小应用程序)：从Web层接收的一个Web页面可以包含内嵌的applet，需要运行在客户端安装了Java虚拟机的Web浏览器上；
- 👉 客户端应用程序：运行在客户机上，能提供强大而灵活易用的用户界面。应用程序可直接访问运行在业务层的EJB，如果需求允许，也可以打开HTTP连接来建立与运行在Web层上的Servlet之间的通讯。



## 1、J2EE组件

### Web组件

- ☞ 既可以是servlet也可以是JSP页面；
- ☞ Servlet是一个Java类，它可以动态地处理请求并作出响应；
- ☞ JSP页面是一个基于文本的文档，它以servlet的方式执行，但是它可以更方便建立动态内容。



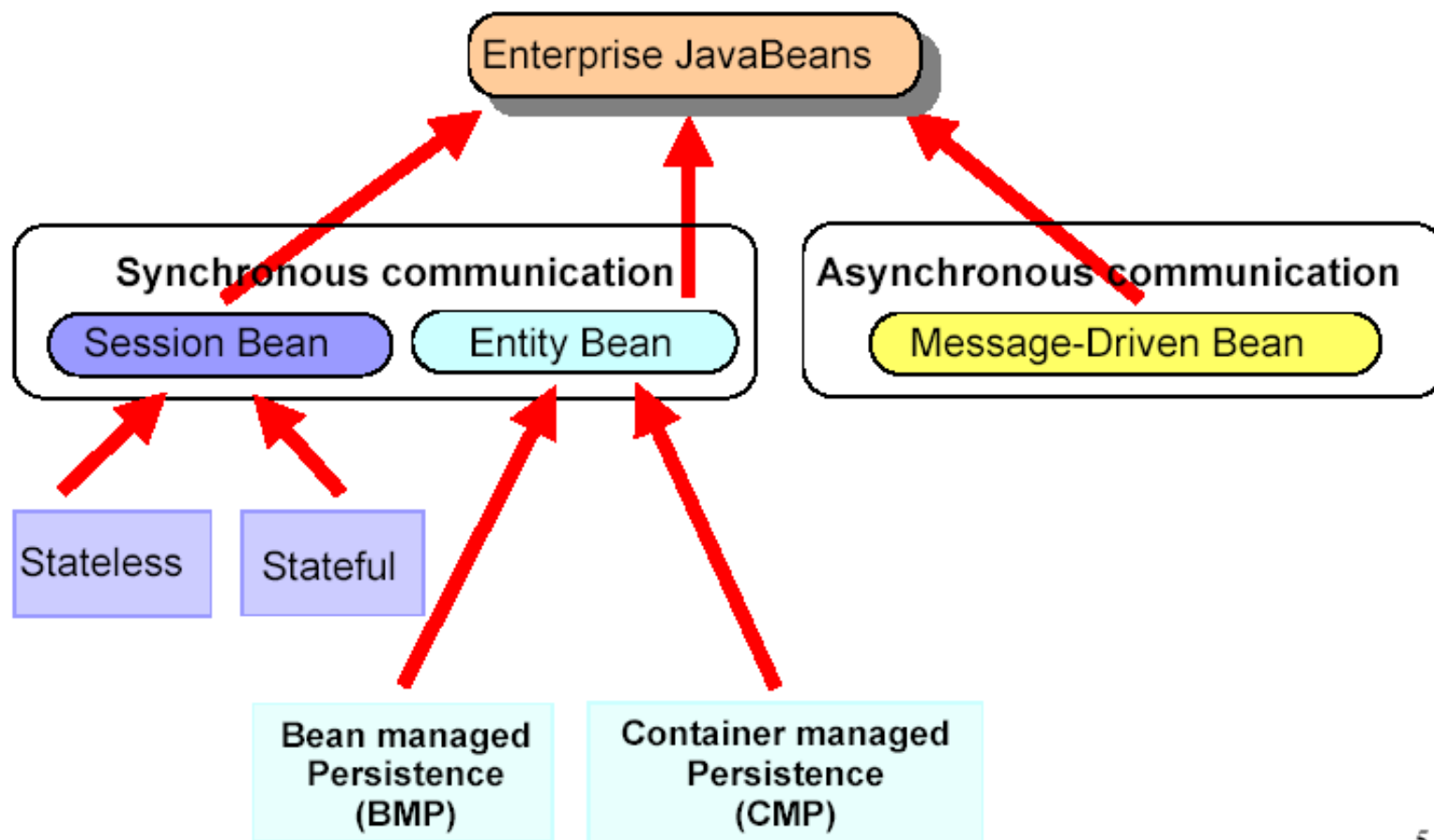
## 1、J2EE组件

### 业务组件

- 👉 J2EE 将业务逻辑从客户端软件中抽取出来，封装在EJB（Enterprise JavaBean）组件中。客户端软件通过网络调用组件提供的服务以实现业务逻辑，而客户端软件的功能单纯到只负责发送调用请求和显示处理结果。
- 👉 有三种类型的Enterprise JavaBeans (EJB)：会话EJB，实体EJB，消息驱动EJB。



## 1、J2EE组件





## 1、J2EE组件

### 企业信息系统层（EIS）

- ❏ 企业信息系统层处理企业信息系统软件并包含诸如企业资源计划(ERP)、主机事务处理、数据库系统和其它遗留下来的信息系统；
- ❏ J2EE应用程序组件可能需要访问企业信息系统，例如获得一个数据库连接。



# J2EE

---

## 2、J2EE容器

**J2EE**服务器以容器的形式为每一个组件类型提供底层服务（如事务处理、状态管理、多线程、资源池等），因此不需要自己开发这些服务，从而使我们可以全力以赴地着手处理业务问题。

**EJB**容器

**Web**容器

客户端应用程序容器

**Applet**容器



## 3、Java Servlet技术

Servlet是用Java编写的Server端程序，与协议和平台无关，采用请求—响应模式提供Web服务,可完成以下任务：

获取客户端浏览器通过HTML表单提交的数据及相关信息

创建并返回对客户端的动态响应页面





访问服务器端资源，如文件、数据库

为JSP页面准备动态数据，与JSP一起协作创建响应的页面





## 4、Java Servlet特点

-  Web浏览器并不直接和servlet通信，servlet是由Web服务器加载和执行的；
-  servlet是用Java编写的，所以一开始就是平台无关的，编写一次就可以在任何平台运行；
-  servlet是持久的，只需Web服务器加载一次，而且可以在不同请求之间保持服务(例如一次数据库连接)；
-  为JSP页面准备动态数据，与JSP一起协作创建响应的页面。



## 5、Java Server Pages (JSP) 技术

JSP技术在传统的网页HTML文件中插入Java程序段和JSP标记，从而形成JSP文件，进而生成页面上的动态内容。

- 👉 简化了Java和Servlet的使用难度，同时通过扩展JSP标记(TAG)提供了网页动态执行的能力。
- 👉 仍未超出Java和Servlet的范围，不仅JSP页面上可直接写Java代码，而且JSP是先被译成Servlet之后才实际运行的。



## 6、Servlet和JSP关系



**JSP**修改后可以立即看到结果，不需要手工编译，**JSP**引擎会来做这些工作；而**Servlet**需要编译，重新启动**Servlet**引擎等一系列动作。



**JSP**提供了一套简单的标签，和**HTML**融合的比较好，可以使不了解**Servlet**的人可以做出动态网页来。对于**Java**语言不熟悉的，会觉得**JSP**开发比较方便。



## 6、Servlet和JSP关系



一个HTML中嵌套Java，一个Java中嵌套HTML



**JSP**



**Servlet**