

Web 开发基础教程 —Html 样式表(Css) Javascript

课程名称:Web 开发基础教程

课程目标:了解 Web 基础知识,熟练应用 Html 常用标签,利用 css 美化页面,学会使用 JavaScript 脚本语言

第一部分 Html 基础知识

第一讲 了解 Html,web 和 internet

了解 *internet* 的起源,浏览器介绍,Html 的简单介绍

第二讲 建立你的第一个 Html 页面

了解 *Html* 文件的构成,标签元素的基本概念,常用的标签,建立你自己的 *Html* 页面

第三讲 设计漂亮的 **Html** 文档 学会使用字符实体,定制页面背景,文字的颜色,字体,文字大小,设置链接的颜色等 第四讲 用好头标签让浏览器识别你的文档 学会使用 `<title>` 标签,了解 `<meta>` 标签设置搜索关键字,字符集等信息

第五讲 格式化页面的正文 怎样设置标题,段落,如何预格式化你的文本,学会使用列表,使用标签修饰你的正文文本. 第六讲 在网页中使用链接 为链接添加辅助信息,利于链接在页面间跳转,指定窗口打开链接,为链接指定快捷键 第七讲 创建有声有色的网页

设置图片尺寸,边框,位置,如何设置图文混排,在网页中插入多媒体,在网页中使用背景音乐

第八讲 用表格展示你的信息 表格的结构介绍,表格中数据的布局,单元格的合并,为表格设置背景,使用嵌套的表格 第九讲 使用框架划分窗口

框架的用途,创建简单的框架,利用框架分割窗口,框架的嵌套,修改框架的显示属性,在框架中使用链接,使用预定义的框架名,浏览器不支持框架的处理

第十讲 利用表单和用户交互

如何建立你自己的表单,学会使用 `<Input>` 标签建立各种表单控件,使用下拉列表和多行 文本框,定制你的按钮

第二部分 用 css 美化你的页面

第十一讲 如何引用样式表

了解样式表的优点,认识样式表,学习如何在页面中引用样式表,如何在样式表中使用注释

第十二讲 字体控制 字体系列,字号,

字体风格,文字修饰 第十三讲 文字

的定位

字间距,行高,文字对齐,缩进,控制边距,边框和空白填充

第十四讲 控制颜色和背景 颜色,背景色,背景图片的添加和控制

第十五讲 要素的定位

绝对定位和相对定位,宽度,高度,可视性控制

第三部分 使用 **JavaScript** 和用户交互

第十六讲 **JavaScript** 语言概况

什么是 *JavaScript*,*JavaScript* 的特点以及与其他语言的区别,*JavaScript* 的运行环境,建立一个

简单的 *JavaScript*

第十七讲 **JavaScript** 基本数据结构

如何嵌入 *JavaScript* 代码,基本的数据类型,变量和表达式

第十八讲 **JavaScript** 程序构成 流程控制语句,函数,事件驱动和事件处理

第十九讲 基于对象的 **JavaScript** 语言

JavaScript 对象的定义和引用,*JavaScript* 的常用对象(字符串,数组,日期等)

第二十讲 系统函数,常用的内部对象

系统函数,常用的内部对象 第二十一

讲 使用框架和 **Cookies**

在 *JavaScript* 中使用框架,设定 *cookies*,删除 *cookies*

第二十二讲用 **JavaScript** 建立日历控件

综合应用 *JavaScript* 建立日历控件

第四部分 综合应用:建立你自己的网站

第二十三讲 建立你自己的网站

目录

第一部分 Html基础知识	8
第一讲:了解Html,web和internet.....	8
1.1 互联网的起源.....	8
1.2 什么是www	8
1.3 web浏览器介绍	8
1.4 超链接.....	9
1.5 Url介绍	10
1.6 Html介绍	10
第二讲 建立你的第一个Html页面.....	12
2.1 Html概述	12
2.2 建立我们的第一Html页面.....	12
2.3 了解Html元素	13
2.4 了解一下Html的标签.....	14
2.5 标签的嵌套.....	16
2.6 综合例子:学会使用标题,段落,折行和水平线,背景	17
第三讲 设计漂亮的Html文档.....	18
3.1 标签的缩进.....	19
3.2 Html的空格规则	19
3.3 html中使用注释	20
3.4 在Html中使用字符实体.....	20
3.5 使用body属性定制丰富多彩的页面.....	22
3.5.2 用bgcolor控制页面的背景颜色.....	22
3.5.3 使用背景图.....	23
3.5.4 使用文字颜色.....	24
3.5.5 改变链接的颜色.....	25
3.6 使用Font标签定制字体	26
3.7 建立一个综合的页面.....	27
第四讲 用好头标签让浏览器识别你的文档.....	30
4.1 DOCTYPE的使用	30
4.2 理解<head>元素.....	30
4.2.1 使用<title></title>标签	30
4.2.2 <meta>标签	30
4.2.3 <style>标签.....	31
4.2.4 <script>标签	31
第五讲 格式化页面的正文.....	32
5.1 标题<h1>到<h6>	32
5.2 <p></p>段落标签.....	32
5.3 <form></form>表单标签	32
5.4 <table></table>表格标签	32
5.5 <hr>水平线标签.....	33
5.6 <div>分段标签	34
5.7 <pre></pre>预格式化标签	34
5.8 使用列表标签.....	35

5.8.1 使用无序列表.....	35
5.8.2 使用有序列表.....	37
5.8.3 使用定义列表定义术语.....	40
5.9 使用链接.....	40
5.10 修饰文字.....	40
第六讲 在网页中使用链接.....	42
6.1 为链接弹出辅助信息.....	42
6.2 使用Name属性在文本内作标签.....	42
6.3 使用target属性.....	42
6.4 为链接指定快捷键.....	42
6.5 链接的综合例子.....	42
第七讲 创建有声有色的网页.....	45
7.1 在网页中使用图片.....	45
7.1.1 图像的对齐方式.....	45
7.1.2 使用width,height设置图片的尺寸.....	45
7.1.3 为图片设置边框.....	45
7.1.4 使用hspace,vspace调整图片的位置.....	46
7.2 使用图像作为链接.....	46
7.3 常用图片格式介绍.....	46
7.4 使用图片的综合例子.....	46
7.5 使用链接嵌入多媒体.....	48
第八讲 用表格展示你的信息.....	50
8.1 表格简介.....	50
8.2 使用标签表示html中的表.....	50
8.2.1 设定表格的边框.....	52
8.2.2 设定表格的对齐方式.....	52
8.2.3 设定表格和单元格的宽度和高度.....	52
8.2.4 控制单元格距离.....	52
8.2.5 表格中数据的布局.....	52
8.2.6 为表格设置背景颜色.....	53
8.2.7 禁止单元格中的文本折行.....	53
8.2.8 合并单元格.....	53
8.2.9 表格的嵌套.....	53
8.2.10 为表格和单元格加入背景.....	53
8.2.11 使用在width中使用"*".....	54
8.2.12 表格各个属性设置应用.....	54
第九讲 使用框架划分窗口.....	65
9.1 框架简介.....	65
9.2 创建frameset文档.....	65
9.3 <frameset>的用法.....	66
9.4 使用cols属性垂直分割窗口.....	66
9.5 使用rows属性水平分割窗口.....	66
9.6 同时应用cols和rows.....	66
9.7 frameset的嵌套.....	68

9.8 设置边框颜色.....	68
9.9 在frame中插入文档.....	69
9.9.1 使用frameborder设置是否显示边框.....	69
9.9.2 marginheight,marginwidth设置页面在框架中的位置.....	70
9.9.3 控制滚动条,取消框架尺寸大小调整.....	70
9.9.4 name属性.....	71
9.10 在框架中使用链接.....	71
9.10.1 使用<base>标签设定缺省的目标框架.....	73
9.10.2 预定义的目标框架名.....	73
9.11 <noframes>为不支持框架的浏览器提供替代内容.....	75
第十讲 利用表单和用户交互.....	76
10.1 表单介绍.....	77
10.2 建立一个最简单的表单.....	77
10.3 <form>标签.....	77
10.4 表单控件标签的用法.....	78
10.4.1 <input>标签的使用.....	78
10.4.2 input标签的综合例子.....	80
10.4.3 下拉框.....	81
10.4.4 多行文本框.....	82
10.4.5 使用button创建按钮.....	82
10.4.6 readonly和disable属性.....	82
10.4.7 设置表单顺序和快捷键.....	82
10.5 表单的处理.....	82
10.6 表单综合例子.....	83
第二部分 用css美化你的页面.....	86
第十一讲 如何引用样式表.....	86
11.1 样式表概述.....	86
11.2 建立第一张样式表.....	87
11.3 把样式表加入到网页中.....	88
11.4 分类引用.....	90
11.5 注释.....	91
第十二讲 字体控制.....	92
12.1 按名称调用字体.....	92
12.2 字号的控制.....	92
12.3 字体风格.....	93
12.4 文字修饰 (text-decoration).....	93
12.5 应用举例.....	93
第十三讲 文字的定位.....	95
13.1 字间距.....	95
13.2 行高.....	95
13.3 文字对齐.....	95
13.4 文字缩行.....	96
13.5 控制边距,边框和空白填充.....	96
13.6 综合例子.....	97

第十四讲 控制颜色和背景.....	100
14.1 颜色.....	100
14.2 背景色.....	100
14.3 背景图片.....	100
14.4 控制背景图片.....	100
14.5 综合例子.....	101
第十五讲 要素的定位.....	105
15.1 绝对定位.....	105
15.2 相对定位.....	105
15.3 控制要素的宽度.....	105
15.4 控制高度.....	105
15.5 可视性.....	105
第三部分 使用JavaScript和用户交互.....	107
第十六讲 JavaScript语言概况.....	108
16.1 什么是JavaScript.....	108
16.2 JavaScript和Java的区别.....	109
16.3 JavaScript程序运行环境.....	110
16.4 编写第一个JavaScript程序.....	110
第十七讲 JavaScript基本数据结构.....	112
17.1 JavaScript代码的加入.....	112
17.3 基本数据类型.....	114
17.4 变量.....	115
17.5 表达式和运算符.....	115
17.6 综合例子.....	116
第十八讲 JavaScript程序构成.....	118
18.1 流程控制语句.....	119
18.2 函数.....	123
18.3 事件驱动及事件处理.....	126
第十九讲 基于对象的JavaScript语言.....	128
19.1 Javascript对象的基础知识.....	129
19.2 JavaScript的常用对象.....	131
19.3 综合例子.....	141
第二十讲 系统函数,常用的内部对象.....	144
20.1 系统函数.....	144
20.2 常用的内部对象.....	145
20.2.1 内部对象树.....	145
20.2.2 navigator 浏览器对象.....	146
20.2.3 screen 屏幕对象.....	147
20.2.4 window 窗口对象.....	147
20.2.5 history 历史对象.....	149
20.2.6 location 地址对象.....	150
20.2.7 document 文档对象.....	151
20.2.8 anchors[]; links[]; link 连接对象.....	152
20.2.9 embeds[] 插件对象.....	152

20.2.10 forms[]; Form 表单对象.....	153
20.3 综合例子.....	161
第二十一讲 使用框架和Cookies.....	164
21.1 使用框架.....	164
21.2 使用 Cookies.....	164
21.3 使用cookie 的例子.....	165
第二十二讲 用JavaScript建立日历控件.....	167
22.1 引用日历控件的html.....	167
22.2 日历控件的js代码.....	168
第四部分 综合应用:建立你自己的网站.....	177
第二十三 讲建立你自己的网站.....	177
23.1 准备工作.....	177
23.2 网站目录结构.....	177
23.3 制做主页.....	178
23.4 制做相识雨轩栏目.....	179
23.5 制做轻松一刻栏目.....	180
23.6 制做雁过留声栏目.....	182
23.7 制做美文欣赏栏目.....	184
23.8 制做顶部的LOGO菜单.....	186
23.9 左侧菜单.....	187
23.10 首页内容.....	189
23.11 友情连接.....	190
23.12 版权信息.....	192
23.13 制做样式表.....	193
23.14 网站的上传.....	194
23.15 课程总结.....	194

第一部分 Html 基础知识

在这一部分我们将学习:web 概述,Html 标签的使用,如何发表自己的 web 页面,使用工具编辑自己的 Web 页面

第一讲:了解 Html,web 和 internet

Internet 是为在计算机之间交换信息而构成的国际网络,万维网(www)是巨大的信息,艺术,商业互联库.

Html(Hypertext Markup language,超文本标记语言)是一种用户制作网页的计算机语言

1.1 互联网的起源

Internet 起源于 60 年代中期美国国防部下属的 DARPA(Defense Advanced Research Project)的网络计划,科学家和科研人员开发 internet 以便于不同的研究机构间通信和共享信息.为了实现这一点,他们创建了新的协议以使计算机之间能够交流信息.现在的 internet 就是建立的这些协议的基础上的.

随着互联网的研究机构,大学,和高新技术公司的逐渐增多,Internet 协议变得越来越流行,这些协议允许你发送 Email,利用 FTP 交换文件,利用 bbs 阅读帖子等等.

Email:电子邮件通过 smtp(简单邮件传输协议)从一个用户发送给另外一个用户.Email 可以说是互联网上最流行的应用.

FTP(文件传输协议):网络上交换文件的一种方法,可以把文件放在 FTP 文件服务器上,你可以利用客户端程序获取这些文件.

TCP/IP(传输控制协议/网间协议)在网络和计算机之间传输信息.

1.2 什么是 www

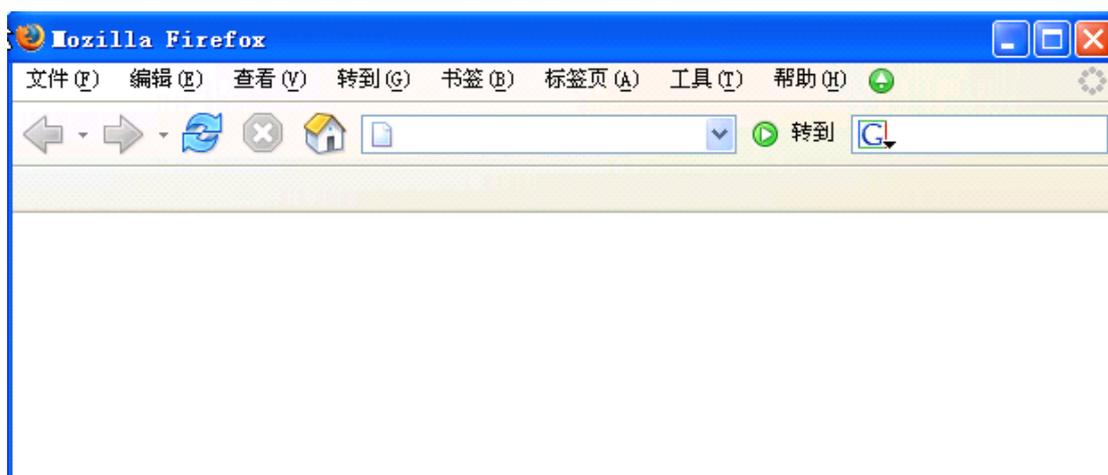
1990 年一种新的被称为 Http(超文本传输协议)出现,www(World Wide Web)也随之诞生,World Wide Web 也简称 W3 或 www 或 Web,是基于 Http 协议的巨大网络.

1.3 web 浏览器介绍

Web 浏览器是浏览 web 的客户端程序,最流行的 Web 浏览器包括 Netscape 的 Navigator,Microsoft 的 IE 浏览器,包括最近很火的 FireFox 浏览器,以及腾讯的 TT 等等
IE 浏览器:



Firefox 浏览器:



1.4 超链接

位于不同服务器上的文件可以通过超链接组织起来,在同一个文件中也可以通过超链接在文件的不同部分之间进行跳转.超链接可以指向文本,图像,音频,视频等媒体文件,在网页上文本的超链接是高亮显示的,并且有下划线,用户可以点击,链接的指定资源.

有了超链接,Web 就成为一个难以想象的信息网络,有了超链接你就可以链接到你感兴趣的资源.



网页 图片 新闻 论坛 网页目录 更多 »

搜索所有网页 搜索所有中文网页 搜索简体中文网页

高级搜索
使用偏好
语言工具

广告计划 - Google 大全 - Google.com in English

©2005 Google - 搜索 8,058,044,651 张网页

1.5 Url 介绍

Url:统一资源定位器,简单的理解为网页或 internet 上的其它文件的地址,Web 上的每个 Url 都是唯一的.

简单的Url构成: <http://www.szonline.net/> 这是深圳热线的网站的Url,它包括web协议(http),后面跟"."之后是斜杠"/"之后是网站的域名.域名是大小写不敏感的.

带ip地址的Url:<http://201.122.122.3> 这组数字给出了互联网上的某台机器的Ip地址,他和使用域名访问是等价的,实际上我们输入域名后,也是要经过域名服务器(DNS)解析成Ip地址后进行访问的.使用域名是为了方便记忆.

带路径和文件名的 Url:Url 也可以带有路径和文件名,如 :
<http://tech.szonline.net/Channel/content/2005/200508/20050803/382283.html>

带有端口号的 Url: <http://lxz003.cn:1088>

具有ftp协议的Url: <ftp://ftp.lxz003.cn/pub>

Url 的格式: 协议:地址

对于 Http 协议的 url 格式如下:<http://网站名:端口号/目录/文件>

其它的协议还有:

Url 中使用的协议列表:

协议关键字	描述
ftp://	文件传输协议
http://	超文本传输协议
file://	本地文件

注意:url 中是不允许有空格的.

1.6 Html 介绍

Html:超文本标记语言 超文本是指定义了额外特征的普通文本,如格式,图像,多媒体,其它链接等. 标记语言是一种特殊类型的计算机语言,因为他只关心对文件中的个部分按其功能分类,表明

哪一部分是文章的标题,哪一部分是子标题等等.

Html 是把用户的文件按各部分进行分类,以便浏览器可以正确的显示它.

Html 标准是由 W3C 负责的,我们可以访问他们的网站:<http://www.w3.org>,Html 是 SGML 标准通用标记语言家族中的一员,Html 的第一版本是 Html1.0,之后又经历了 Html+,Html2.0,Html3.0,现在发展到了 Html4.0,日趋完善和强大.



第二讲 建立你的第一个 Html 页面

Html 可以说是简单而且丰富多彩的,利用简单的标签就可以向你展示一个奇妙的世界,我们在这一讲将了解 Html 的结构,学习简单的标签,之后组合这些标签向你展示一篇优美的散文.

2.1 Html 概述

HTML 表示超文本标记语言 (Hyper Text Markup Language)。
HTML 文件是一个包含标记的文本文件。这些标记保速浏览器怎样显示这个页面。
HTML 文件必须有 htm 或者 html 扩展名。
HTML 文件可以用一个简单的文本编辑器创建。

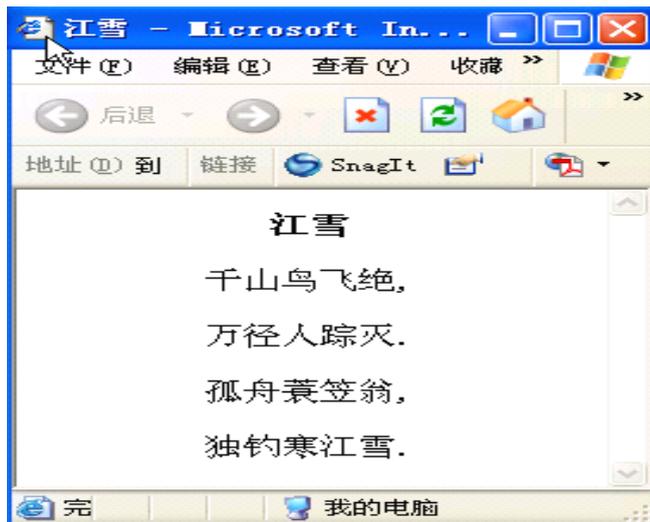
2.2 建立我们的第一 Html 页面

建立 Html 很简单,打开文本编辑器,输入下面的代码保存为 snow.html 双击它我们就可以在浏览器中看到我们的第一个 Html 页面了.

代码:

```
<html>
  <head>
    <title>江雪</title>
  </head>
  <body>
    <center><b>江雪</b></center><br>
    <center>千山鸟飞绝,</center><br>
    <center>万径人踪灭.</center><br>
    <center>孤舟蓑笠翁,</center><br>
    <center>独钓寒江雪.</center>
  </body>
</html>
```

在浏览器中显示:



我想看一下:[htmlCode\snow.HTML](#)

看看我们的第一个 Html 页面是怎样构成的:

简单的讲一个标准的 Html 页面是由两个部分构成的,<Head></Head>构成头部,由<body></body>构成身体部分.

在 HTML 文档中,第一个标签是<html>.这个标签告诉浏览器这是 HTML 文档的开始。HTML 文档的最后一个标签是</html>,这个标签告诉浏览器这是 HTML 文档的终止.在<head>和</head>标签之间文本的是头信息。在浏览器窗口中,头信息是不被显示的。在<title>和</title>标签之间的文本是文档标题,它被显示在浏览器窗口的标题栏。

爬楼和人生 - Microsoft Internet Explorer

如果我们没有在 Html 中指定 Title,在标题的部分回显示当前文件的路径信息.

在<body>和</body>标签之间的文本是正文,会被显示在浏览器中.在和标签之间的文本会以加粗字体显示.

<center>和</center>标签之间的文本将居中显示.

表示折行显示,它没有对应的</br>标签是独立的标签

注意:

- z 为了使我们的 **Html** 代码更清晰,便于阅读和维护,我们不要漏掉结束标签,不同层次嵌套的标签要有一定的缩进.让我们的代码也有一种诗一样的韵律感.
- z 当我们修改了 **Html** 代码时,我们只要点击浏览器中的刷新按钮就可以看到改动之后的效果.
- z 我们也可以在浏览器显示的页面中单击右键,选择查看源文件来查看 **Html** 页面的源代码
- z 你也可以使用可视化的网页制作工具比如 **frontpage**, **dreamwaver** 来创建你自己的网页,但从熟悉 **Html** 代码的角度,我们不提倡这样做.

2.3 了解 Html 元素

HTML 文档是由 HTML 元素组成的文本文件.

HTML 元素:

包含预定义的正在使用的 HTML 标签,和应用标签的内容, 江雪即算是一个 Html 元素,同时我们也可以把我们的第一个 Html 页面中的<body>.....</body>部分看成是一个元

素.

HTML 标签:

HTML 标签用来组成 HTML 元素。HTML 标签两端有两个包括字符: "<" 和 ">" ,这两个包括字符被称为角括号。

HTML 标签通常成对出现,比如和。一对标签的前面一个是开始标签,第二个是结束标签,在开始和结束标签之间的文本是元素内容。

HTML 标签是大小写无关的,跟表示的意思是一样的,但是我们提倡使用小写的标签。

标签属性:

标签是可以拥有属性的。属性能够为页面上的 HTML 元素提供附加信息。例如:标签<body>定义了 HTML 页面的主体元素。使用一个附加的 bgcolor(背景颜色)属性,你可以告诉浏览器,你页面的背景色可以使用蓝色,你可以写下这样的代码:

```
<body bgcolor="blue">
```

同样,我们使用标签<table>可以定义一个 HTML 表格,使用它的一个附加属性 border(边界),你可以向浏览器说明这个表格是没有边框的,我们可以写下如下的代码:

```
<table border="0">
```

属性通常由属性名和值成对出现,就像这样: name="value"。属性通常是附加在 HTML 元素的开始标签上的。

引号的使用:

属性值是包含在引号中的,我们可以使用双引号,但是也可以使用单引号,比如在有些情况下,属性值中本身包含引号,我们就可以使用单引号,比如: name='lxz003 say "Hello world",now the day is fine!'。注意:我们在使用引号时要注意使用半角的符号,而且注意引号要成对出现的。

2.4 了解一下 Html 的标签

<body></body>主体标签:要显示到浏览器中的主体文本。

代码:

```
<html>
```

```
  <body>
```

```
    看到了吗,这里在演示主体标签的用法,它将在浏览器中显示!
```

```
  </body>
```

```
</html>
```

我想看一下:<htmlCode/body.HTML>



看到了吗,这里在演示主体标签的用法,它将在浏览器中显示!

<p></p>段落标签:在浏览器中显示段落,每一段落后会空一行。

代码:

```
<html>
```

```

<body>
  <p>这是第一段.</p>
  <p>这是第二段.</p>
  <p>这是第三段.</p>
  <p>这是第四段.</p>
</body>
</html>

```

我想看一下:[htmlCode\段落标记.HTML](#)

这是第一段.

这是第二段.

这是第三段.

这是第四段.

<h1></h1>...<h6></h6>标题标签:标题显示标签从 **h1** 到 **h6** 逐渐减小字号, 每一个标题标签后面系统会自动跟一个空行.

代码:

```

<html>
  <body>
    <h1>一号标题</h1>
    <h2>二号标题</h2>
    <h3>三号标题</h3>
    <h4>四号标题</h4>
    <h5>五号标题</h5>
    <h6>六号标题</h6>
  </body>
</html>

```

我想看一下:[htmlCode\标题标记.HTML](#)

一号标题

二号标题

三号标题

四号标题

五号标题

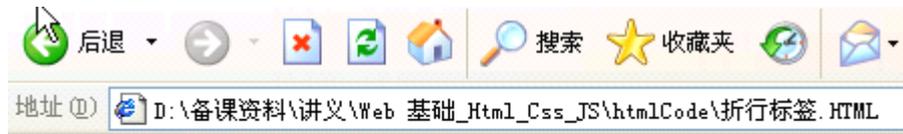
六号标题

**
换行标签:结束当前行,强制换行,它是一个空标签,没有结束标签,不要使用</br>**

代码:

```
<html>
  <body>
    我们要学会使用强制换行标签,<br>看到了吗,强制换行标签在这里已经折行显示了.
  </body>
</html>
```

我想看一下:[htmlCode\折行标签.HTML](#)



我们要学会使用强制换行标签,
看到了吗,强制换行标签在这里已经折行显示了.

<!-- -->注释标签: 注释标签用来在 **HTML** 源文件中插入注释。注释会被浏览器忽略。你可以使用注释来解释你的代码, 这样在你以后的维护很重要。

代码:

```
<html>
  <head>
    <title>江雪</title>
  </head>
  <body>
    <!-- 下面是一首诗 -->
    <center><b>江雪</b></center><br>
    <center>千山鸟飞绝,</center><br>
    <center>万径人踪灭.</center><br>
    <center>孤舟蓑笠翁,</center><br>
    <center>独钓寒江雪.</center>
  </body>
</html> 我想看一
```

下:[htmlCode\snowComment.HTML](#) 注意:

- z 注释标签只有一个!号,注释信息不会显示在浏览器中.
- z 我们也不要期望使用空格,来进行排版,浏览器对多个空格会当一个空格来处理
- z 我们也不建议使用<p></p>来插入空行,我们要使用

去插入空行.<p>可以不使用</p>标记,但我们还是尽量保持标签的对应性.

2.5 标签的嵌套

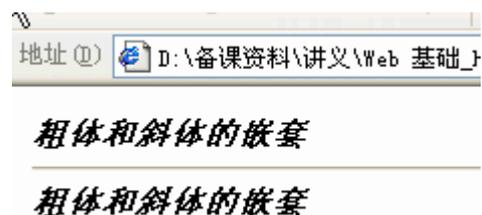
我们在使用 Html 标签的时候是可以嵌套的,任何正文信息都是嵌套在<body></body>标签中,同样<body></body>标签又嵌套在<html></html>标签中,其它的嵌套的例子比如:

<i>加粗和斜体的嵌套</i>

或者可以:

<i>加粗和斜体的嵌套</i>

我想看一下:[htmlCode\Html标签的嵌套.HTML](#)



嵌套时要注意它们的嵌套顺序嵌套要互相匹配,比如<i>不正确的嵌套</i>.

2.6 综合例子:学会使用标题,段落,折行和水平线,背景

代码:

```
<html>
  <head>
    <title>爬楼和人生</title>
  </head>
  <body bgcolor="blue">
    <h3>爬楼和人生</h3>
    <hr>
    <p>
      有一对兄弟,有一天他们出去爬山,然后一起回家。他们的家住在八十层楼,他们一人背着一大包的行李回家,却发现大楼停电了。于是哥哥就说:弟弟,我们一起爬楼梯上去吧。
      于是他们就一起爬上去。
      到了二十楼的时候,哥哥
      又告诉弟弟:
      包包太重了,我们把它放在二十楼,我们爬上去,明天再下来拿。
      弟弟说:好。
      于是他们就把他们的包包放在二十楼,继续往上爬。到了四十楼,弟弟开始抱怨,于是就跟哥哥吵起来了。他们边吵边爬,爬到了六十楼,哥哥就对弟弟说:只剩二十层楼了,我们不要吵了,默默的爬完它吧!!于是他们就各走各的,终于到了家门口。哥哥就摆出了很帅的姿势:弟弟开门。
      弟弟就对哥哥说:别闹了,钥匙不是在你那儿。
      结果,他们把钥匙留在二十楼的包包里了。
    </p>
    <p>这个故事其实在反映我们的人生,有很多人在二十岁以前是活在家人的期望和老师的期许之下,背负着很多的压力;在二十岁之后离开了众人的压力,开始满腔的热血,开始有很多的梦想去完成;
```

可是工作了二十年后, 开始发觉工作不如意...
 于是就开始抱怨老板、抱怨公司、抱怨社会、抱怨政府。就在这抱怨中又渡过了二十年。于是告诉自己, 六十岁了没什么好抱怨的了, 就默默的走完自己的这些年吧。到了八十岁快要死掉的前夕, 才想起自己好像有什么事还没完成
 原来, 他二十岁的梦想还没有完成。

</p>

<p> 所以, 希望现在的你充满着理想...充满着热情去做自己想做的事。
 人生短短数十载, 也没有什么好担心的不是吗? 尤其是在经历那么多的遭遇, 更应该能体会, 把握现在。

 记得吗? 今天就是礼物!

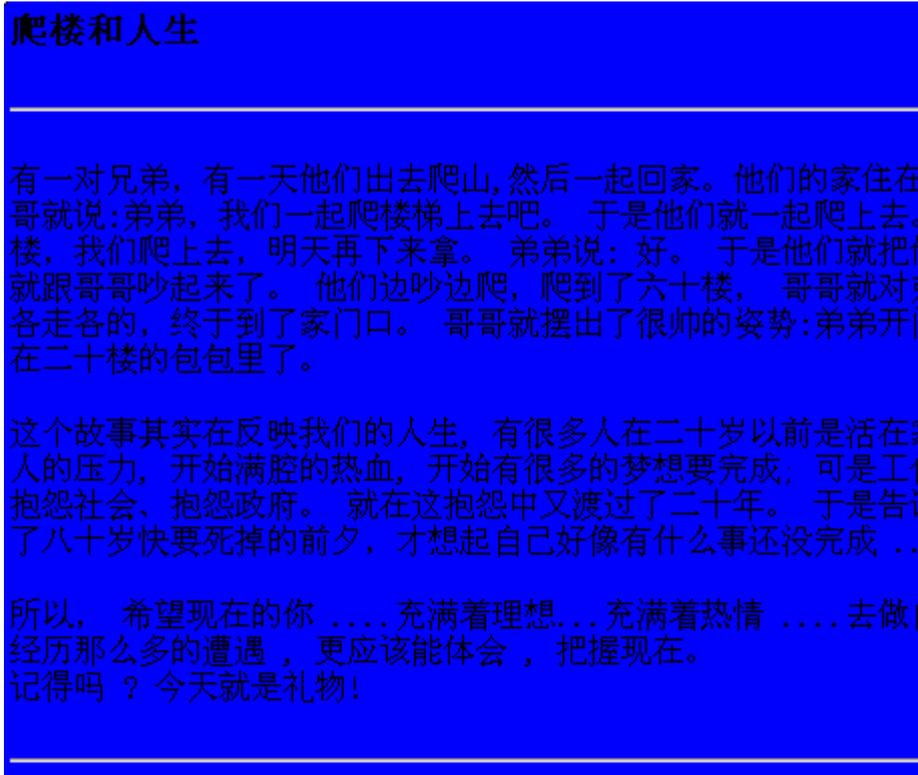
</p>

<hr>

</body>

</html>

我们在这里使用了标题,水平线,设置文档的背景色等标记和属性
 我来看一下:[htmlCode\爬楼和人生.HTML](#)



通过以上的介绍我们对 Html 已经有了感性的认识, 下面我们就要开始我们真正的 Html 学习之旅了。

第三讲 设计漂亮的 Html 文档

上一讲我们讲述了 Html 文档的基本结构, 简单常用的标签, 并且利用 Html 展示了一篇优

美的文章.在接下来的课程中我们将向你讲述如何组合 Html 的标签展示漂亮的 html 文档,在这一讲你将学会: 设定文档的背景 改变文字的颜色,字体大小,字体风格 设置链接的颜色
综合应用各种标签

3.1 标签的缩进

我们在书写 Html 时,我们也提倡像写代码一样进行缩进,利用缩进我们可以创造出像诗一样有韵律的代码,增加代码的美感,采用缩进的原则,也有利于代码的阅读和维护.你可以在严格缩进的代码中迅速找出不匹配的标签等错误.杂乱无章的代码,我们也没有心情去修改和维护.

我们在对代码进行缩进时,一般要缩进 4 个空格,我们也可以使用 tab 键去缩进代码.我们可以回头看一下”爬楼与人生.html”的源码,采用缩进的写法后是不是更像一首诗呢,如果你不信我把这段代码打乱给你看,看看你有多大的耐心把它读完:

爬楼和人生.html

```
<html><head><title>爬楼和人生</title></head><body bgcolor="blue"><h3>爬楼和人生</h3><hr><p>有一对兄弟,有一天他们出去爬山,然后一起回家。他们的家住在八十层楼他们一人背着一大包的行李回家,却发现大楼停电了。于是哥哥就说:弟弟,我们一起爬楼梯上去吧。于是他们就一起爬上去。到了二十楼的时候,哥哥又告诉弟弟:包包太重了,我们把它放在二十楼,我们爬上去,明天再下来拿。弟弟说:好。于是他们就把他们的包包放在二十楼,继续往上爬。到了四十楼,弟弟开始抱怨,于是就跟哥哥吵起来了。他们边吵边爬,爬到了六十楼,哥哥就对弟弟说:只剩二十层楼了,我们不要吵了,默默的爬完它吧!!于是他们就各走各的,终于到了家门口。哥哥就摆出了很帅的姿势:弟弟开门。弟弟就对哥哥说:别闹了,钥匙不是在你那儿。结果,他们把钥匙留在二十楼的包包里了。</p><p>这个故事其实在反映我们的人生,有很多人在二十岁以前是活在家人的期望和老师的期许之下,背负着很多的压力;在二十岁之后离开了众人的压力,开始满腔的热血,开始有很多的梦想要完成;可是工作了二十年后,开始发觉工作不如意...于是就开始抱怨老板、抱怨公司、抱怨社会、抱怨政府。就在这抱怨中又渡过了二十年。于是告诉自己,六十岁了没什么好抱怨的了,就默默的走完自己的这些年吧。到了八十岁快要死掉的前夕,才想起自己好像有什么事还没完成.....原来,他二十岁的梦想还没有完成。</p><p>所以,希望现在的你....充满着理想...充满着热情....去做自己想做的事。人生短短数十载,也没有什么好担心的不是吗?尤其是在经历那么多的遭遇,更应该能体会,把握现在。<br>记得吗?今天就是礼物!</p><hr></body></html>
```

3.2 Html 的空格规则

在 Html 中的空格规则为:多个空格,制表符,回车将转换为单一的空格.我们看一下含有空格的源码和显示后的结果进行对比:

代码:

```
<html>
  <head><title>在 Html 中使用空格</title></head>
```

```

<body>
  <h3>在 Html    中使用空格

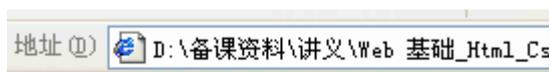
  </h3>
  <p>

      1
    1   1
  1     1
1       1
  1     1
    1   1
      1

  </p>
</body>

```

</html> 我们的运行结果会忽略这些空格. 我想看一下:[htmlCode\在Html中使用空格.HTML](#)



在Html 中使用空格

```
1 1 1 1 1 1 1 1 1 1 1 1
```

我们也可以使用预处理格式标签绕过 Html 的空白规则.

3.3 html 中使用注释

注释的使用可以方便我们理解 Html 的代码,Html 的注释标签是<!-- -->,它的使用方法我们已经在第二讲中进行了讲解.

3.4 在 Html 中使用字符实体

在 HTML 中,有些字符拥有特殊含义,比如小于号 "<" 定义为一个 HTML 标签的开始.假如我们想要浏览器显示这些字符的话,必须在 HTML 代码中插入字符实体.

一个字符实体拥有三个部分:一个 and 符号 (&),一个实体名或者一个实体号,最后是一个分号 (;)

想要在 HTML 文档中显示一个小于号,我们必须这样写: <或者<

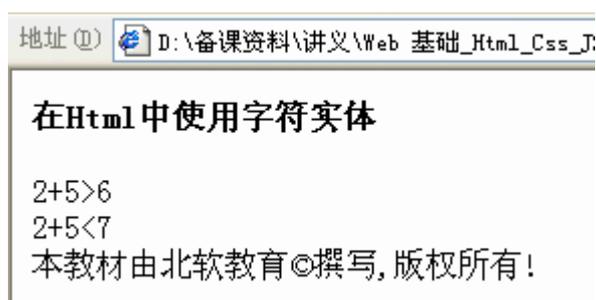
使用名字相对于使用数字的优点是容易记忆,缺点是并非所有的浏览器都支持最新的实体名,但是几乎所有的浏览器都能很好地支持实体号.

注意:实体名是大小写敏感的.

我们使用字符实体的代码:

```
<html>
  <head><title>在 Html 中使用字符实体</title></head>
  <body>
    <h4>在 Html 中使用字符实体</h4>
    2+5&#62;6<br>
    2+5&lt;7 <br>
    本教材由北软教育&copy;撰写,版权所有!
  </body>
</html>
```

我想看一下:[htmlCode\在Html中使用字符实体.HTML](#)



最常用的字符实体:

显示结果	描述	实体名	实体号
	不可分隔空格	 	
<	小于	<	<
>	大于	>	>
&	And 符号	&	&
“	双引号	"	"
‘	单引号		'

其它常用的实体字符:

显示结果	描述	实体名	实体号
¢	分	¢	¢
£	英镑	£	£
¥	人民币元	¥	¥
§	章节	§	§
©	版权	©	©
®	注册	®	®
×	乘号	×	×
÷	除号	÷	÷

如何使用不可拆分空格:

 不可拆分空格:我们在前面已经讲述了空格会被浏览器合并,你输入了 10 个空格只会显示一个空格,如果我们真的需要使用 10 个空格,我们可以使用不可拆分空格.浏览器不会在不可拆分空格处折行,也不会和并空格的个数,如果我们想行单词间插入多个空格,或希望单词显示在同一行时可使用不可拆分空格.

代码:

```
<html>
  <head><title>不可拆分空格</title></head>
```



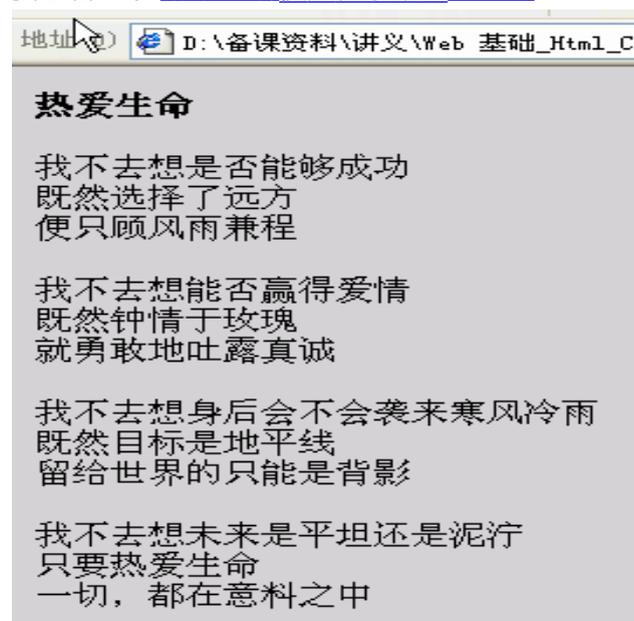
```

便只顾风雨兼程
</p>
<p> 我不去想能否赢得
爱情<br> 既然钟情于玫瑰
<br> 就勇敢地吐露真诚
</p>
<p> 我不去想身后会不会袭来寒风
冷雨<br> 既然目标是地平线<br> 留给
世界的只能是背影
</p>
<p> 我不去想未来是平坦还是泥泞
<br> 只要热爱生命<br> 一切，都
在意料之中
</p>
</body>

```

</html>我们使用 body 的 bgcolor 属性设置 RGB 颜色代码,就可以改变浏览器默认的文档背景颜色.

我想看一下:[htmlCode\使用背景颜色.HTML](#)



3.5.3 使用背景图

我们可以使用背景图美化我们的文档,使用对比度太强的图片作为背景不利于文字的阅读,在 body 中设置 background 属性就可以改变背景图,background 的值使用的是图片的 Url. 代码:

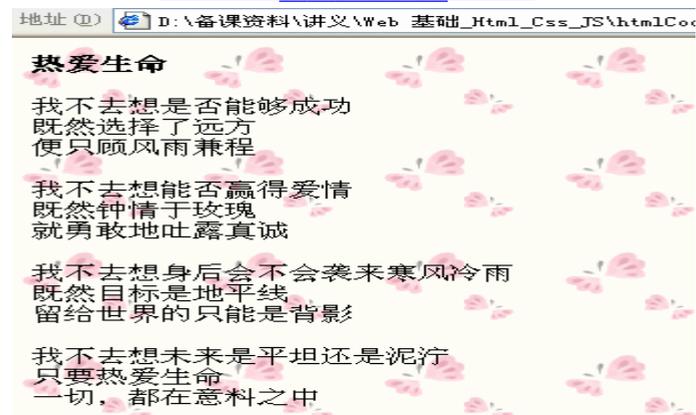
```
<html>
```

```

<head><title>使用背景颜色</title></head>
<body background="butterfly.gif">
  <h4>热爱生命</h4>
  <p> 我不去想是否能够
成功<br> 既然选择了远方
<br> 便只顾风雨兼程
  </p>
  <p> 我不去想能否赢得
爱情<br> 既然钟情于玫瑰
<br> 就勇敢地吐露真诚
  </p>
  <p> 我不去想身后会不会袭来寒风
冷雨<br> 既然目标是地平线<br> 留给
世界的只能是背影
  </p>
  <p> 我不去想未来是平坦还是泥泞
<br> 只要热爱生命<br> 一切，都
在意料之中
  </p>
</body>
</html>

```

我想看一下:[htmlCode\使用背景图片.HTML](#)



3.5.4 使用文字颜色

在 body 中的 text 属性可以改变文字的颜色,浏览器的默认颜色为黑色.

代码:

```

<html>
  <head><title>使用背景颜色</title></head>
  <body background="butterfly.gif" text="#008080">

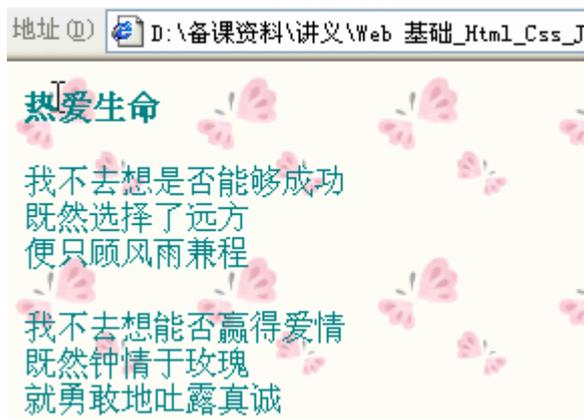
```

```

    <h4>热爱生活</h4>
    <p> 我不去想是否能够
成功<br> 既然选择了远方
<br> 便只顾风雨兼程
    </p>
    <p> 我不去想能否赢得
爱情<br> 既然钟情于玫瑰
<br> 就勇敢地吐露真诚
    </p>
    <p> 我不去想身后会不会袭来寒风
冷雨<br> 既然目标是地平线<br> 留给
世界的只能是背影
    </p>
    <p> 我不去想未来是平坦还是
泥泞<br> 只要热爱生活<br> 一
切，都在意料之中
    </p>
</body>
</html>

```

我想看一下:[htmlCode\改变文字颜色.HTML](#)



3.5.5 改变链接的颜色

Link 设置超文本链接的颜色,IE 默认的颜色为蓝色.如 `<body link="green">`只要未访问的链接就会以绿色显示.

Vlink 设置已访问过的链接颜色,IE 默认的颜色为紫色.如`<body vlink="#800000">`将使用褐红色表示浏览器已访问过的链接颜色.

Alink 激活链接时的颜色,当光标移动到链接上单击的瞬间显示的颜色,IE 默认的颜色为

红色.如<body alink="#FF00FF">单击链接的瞬间为品红色.

代码:

```
<html>
  <head><title>改变链接的颜色</title></head>
  <body link="green" vlink="#800000" alink="#FF00FF">
    <h4>我的收藏</h4>
    <a href="http://www.szonline.net">深圳热线</a><br><br>
    <a href="http://www.google.com">Google</a><br><br>
    <a href="http://www.csdn.net">中国程序员网站</a>
  </body>
</html>
```

我想看一下:[htmlCode\改变链接的颜色.HTML](#)

3.6 使用 Font 标签定制字体

标签允许你定制文字的大小,颜色,字体等.

它的属性有:

color:控制文字的颜色,使用颜色代码或 RGB 码.如生活的理想,就是理想的生活!

size:控制文字的大小,html 设定了 1-7 共七种字号,1 号最小,7 号最大(比 H1 还要大),如要显示比默认字体小一号可以设定 size="-1",比默认字体大一号可以设定 size="+1",我们也可以使用<big>比默认字体大一号</big>或<small>比默认字体小一号</small>这两个标签是嵌套在标签中的子标签.

face:控制文本的字体,设置文字的字体,如设置文字的字体使用它的缺点是,你不能保证你的浏览器支持你指定的字体.

代码:

```
<html>
  <head><title>定制我们的文字样式</title></head>
  <body>
    <font color="blue" face="Arial">既然选择了远方,便只顾风雨兼程!(正常大小)
english</font><br>
    <font color="blue" size="+1" face="Comic Sans MS">既然选择了远方,便只顾风雨兼
程!(大一号) english</font><br>
    <font color="blue" size="-1" face="Courier">既然选择了远方,便只顾风雨兼程!(小一
号) english</font><br>
    <font color="blue" size="1" face="Courier New">既然选择了远方,便只顾风雨兼
程!(1 号) english</font><br>
    <font color="blue" size="2" face="Modern">既然选择了远方,便只顾风雨兼程!(2 号)
english</font><br>
    <font color="blue" size="3" face="MS Sans Serif">既然选择了远方,便只顾风雨兼
程!(3 号) english</font><br>
    <font color="blue" size="4" face="MS Serif">既然选择了远方,便只顾风雨兼程!(4 号)
english</font><br>
    <font color="blue" size="5" face="MS-DOS CP 932">既然选择了远方,便只顾风雨兼
```


地址 (D) D:\备课资料\讲义\Web 基础_Html_Css_JS\htmlCode\跳蚤和爬蚤的故事.HTML

跳蚤和爬蚤的故事

科学家做过一个有趣的实验：他们把跳蚤放在桌上，一拍桌子，跳蚤迅速跳到最高的动物！

然后在跳蚤头上罩一个玻璃罩，再让它跳；这一次跳蚤碰到了玻璃罩。连续保持在罩顶以下高度。接下来逐渐改变玻璃罩的高度，跳蚤都在碰壁后主动改变了。

科学家于是把玻璃罩打开，再拍桌子，跳蚤仍然不会跳，变成“爬蚤”了。一次次受挫学乖了，习惯了，麻木了。最可悲之处就在于，实际上的玻璃罩已经罩在了潜意识里，罩在了心灵上。

行动的欲望和潜能“自我设限”。很多人的遭遇与此极为相似。在成长的过程中特别是幼于是奋发向上的热情、欲望被“自我设限”压制封杀，没有得到及时的疏导与和勇气，渐渐养成了懦弱、犹疑、狭隘、自卑、孤僻、害怕承担责任、不思进表现就是随波逐流。**与生俱来的成功火种过早的熄灭。**

北软教育®选摘
北软网站

第四讲 用好头标签让浏览器识别你的文档

我们在前一讲讲述了如何美化你的 Html 文档,我们接下来的任务是设置头标签让浏览器识别我们的文档,我们在这一讲将了解:

学会使用<title>标签

了解<meta>标签

4.1 DOCTYPE 的使用

DOCTYPE 并不属于 Html 的标签它是 SGML 的一个实体,它优先于 Html 标签,说明 Html 的版本,文档类型定义信息,它一般用于 Html 的最开始处,在网页中也可以不包含 DOCTYPE 信息.我们仅仅描述它的使用,并不过多的说明其它信息.

使用举例:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD></HEAD>
  <BODY></BODY>
</HTML>
```

4.2 理解<head>元素

我们在文档的<head>部分,应该包含文档的标题(<title></title>),索引关键字,语言等信息.<head></head>标签是可选的,在 Html 中可以不包含该标签,浏览器也可以根据实际情况识别文档的头信息.

4.2.1 使用<title></title>标签

Html 文档只允许有一个标题,标题被显示在浏览器的最上方,在<title></title>标签中不能在嵌套其它标签,这样的写法是不对的:<title>文章的标题不要在嵌套标签</title>,但是可以在标题中使用字符实体,比如:<title>北软教育®门户网站</title>,标题的内容不要太长,太长会被浏览器截断,一般要少于 80 个字符.

我们在收藏网页的时候,标题会作为书签收藏到收藏夹中.

4.2.2 <meta>标签

<meta>标签说明文档的自身信息.它不会被浏览器显示给用户,但却给软件程序和搜索引擎提供了关于文档的信息.我们仅了解一下这个标签就足够了.下面是使用这个标签的综合例子:

代码:

```
<!DOCTYPE HTML PUBLIC "-//D3C//DTD HTML 4.0//EN"> -----文档类型,版本信息
<HTML>
```

```

<HEAD>
  <TITLE>使用 meta 标签</TITLE>
  <META NAME =”Author” CONTENT=”文档的作者信息”> ---文档的作者信息
  <META HTTP-EQUIV=”Expires” CONTENT=”Fri ,31 Dec 2005 23:59:59
GMT”>--Http 信息头,传递给服务器的文档附加信息,本行描述网页的过期时间
  <META NAME=”Description” CONTENT=”This doc...”>---文档内容描述
  <META NAME=”Keyword” CONTENT=”music ,ice”> ---检索关键字
  <META NAME=”Copyright” CONTENT=”&copy;2005 by lxz003”>---版权信息
  <META HTTP-EQUIV=”Content-Language” CONTENT=”en”>----Http 信息头,描述
文档的语言信息.
  <META HTTP-EQUIV=”REFRESH” CONTENT=”20”> -Http 每 20 秒刷新一次网页
  <meta http-equiv="content-type" content="text/html; charset=gb2312">--指定字符集
</HEAD>
<BODY>
</BODY>
</HTML>

```

4.2.3 <style>标签

说明文档的显示风格

4.2.4 <script>标签

脚本标签说明使用的脚本语言如:JavaScript,VBScript 等.

第五讲 格式化页面的正文

这一讲我们主要学习怎样设置标题,段落,如何预格式化你的文本,学会使用列表,使用标签修饰你的正文文本.

5.1 标题<h1>到<h6>

标题<h1>到<h6>可以可以使用 align 控制标题的对齐,

<h1 align="left">起点靠左侧对齐

<h1 align="right">起点靠右侧对齐

<h1 align="center">居中对齐

<h1 align="justify">两端对齐(不是所有的浏览器都支持)

我想看一下:[htmlCode\标题标记.HTML](#)

5.2 <p></p>段落标签

<p>可以没有</p>结束标签,在段落标签中也可以使用 align 控制对齐,和上面的标题标签类似:

<p align="left"> 左对齐,浏览器默认

<p align="right">右对齐

<p align="center">居中

<p align="justify">两端对齐 不要使用<p>来产生

空行,实际上空行会被浏览器忽略.

<p>不要用段落标签产生空行</p>

<p>

<p>

<p>

<p>段落的空行被忽略</p>

和下面的代码是等价的.

<p>不要用段落标签产生空行</p>

<p>段落的空行被忽略</p>

5.3 <form></form>表单标签

表单标签是允许用户填入和传输信息的区域.我们会在后面专题详细介绍.

5.4 <table></table>表格标签

表格标签用来生成表格,我们会在后面的专题中详细介绍.

5.5 <hr>水平线标签

<hr>产生水平线,没有结束标签,也可以设置对齐方式和尺寸.

align="left" 左对齐

align="right" 右对齐

align="center" 居中

noshade 实心无阴影的水平线

size="数值" 长度(像素)

width="数值" 宽度(像素)

width="百分比" 宽度(屏幕的百分比)

代码:

```
<html>
  <head><title>使用水平线</title></head>
  <body>
    使用水平线
    <hr>
    既然选择了远方,便只顾风雨兼程!
    <hr noshade>
    既然选择了远方,便只顾风雨兼程!
    <hr size="1">
    既然选择了远方,便只顾风雨兼程!
    <hr size="3">
    既然选择了远方,便只顾风雨兼程!
    <hr size="5">
    既然选择了远方,便只顾风雨兼程!
    <hr size="10">
    既然选择了远方,便只顾风雨兼程!
    <hr size="12" noshade>
    既然选择了远方,便只顾风雨兼程!
    <hr size="5" noshade width="200" align="left">
    既然选择了远方,便只顾风雨兼程!
    <hr size="5" noshade width="200" align="right">
    既然选择了远方,便只顾风雨兼程!
    <hr size="5" noshade width="200" align="center">
    既然选择了远方,便只顾风雨兼程!
    <hr size="5" noshade width="50%" align="center">
    既然选择了远方,便只顾风雨兼程!
    <hr size="5" noshade width="80%" align="center">
  </body>
</html>
```

我想看一下:[htmlCode\使用水平线.HTML](#)



5.6 <div>分段标签

<div>标签可以实现对一段文本进行分段,整体控制它们的对齐方式.div 标签还有很多复杂的行为.

<div>标签也具有对齐属性,包括:

align = "left"

align = "right"

align = "center"

5.7 <pre></pre>预格式化标签

使用预格式化标签可以把预格式的文本包含的页面中,使用预格式化标签可以保持文本中的空白,使用预格式化时尽量使用空格,不使用 tab 键.

代码:

```
<html>
  <head><title>使用预格式化文本</title></head>
  <body>
    <pre>
```

```

      \\\|\\|
      \|  - -  \|
      (  @@  )
oOOo-(_)oOOo
|
|      ,~'~'~_
|      (((\|~_~'~.
|      )a ,~_~',,~'~
|      \_ /,~',,~"
|      20050808      \ )--.
|  公司网站:      \
|      www.octit.cn      )&, ._'
|

```

```

      \ )--.
      Oooo '._.'
  oooO- ( )
  ( ) )/
  \ ( (/
  \)

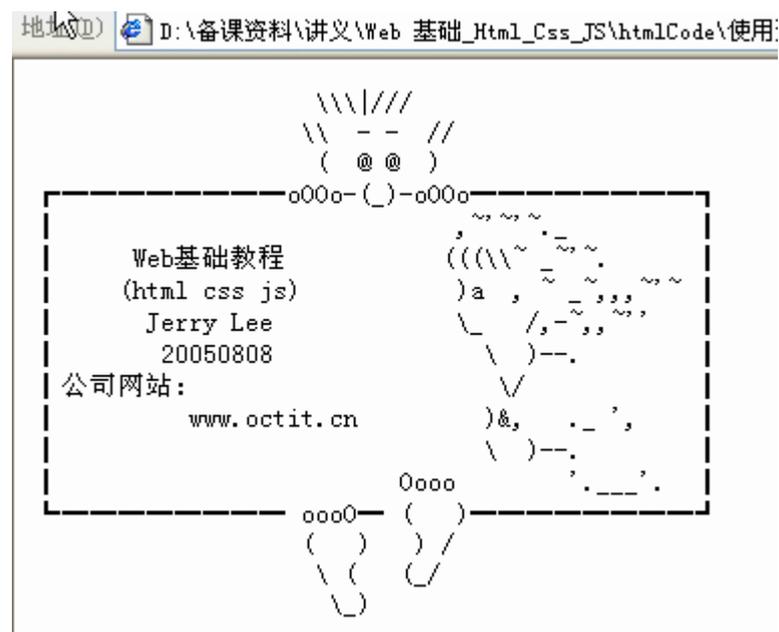
```

</pre>

</body>

</html>

我想看一下:[htmlCode\使用预格式化标签.HTML](#)



5.8 使用列表标签

列表标签分为: 无序列表:用黑点分隔,可以是闭环,开环,方框等多种风格,使用标签. 有序列表: 按编号排序,比如数字,罗马字母,或英文大小写字母,使用标签. 定义列表:包含术语和定义,使用<dl></dl>标签,嵌套术语标签<dt>, 定义标签<dd>.

5.8.1 使用无序列表

使用格式:

```
<ul>
```

```
  <li></li>
```

```
  <li></li>
```

```
</ul>
```

是无序列表标签,嵌套其中表示列表项标签.它们具有相同的 type 属性: type="circle" 空心的圆点

type="disc" 实心的圆点

type="square" 实心方框 列表的默认值为实心圆点. 列表的重要的特定是可以嵌套的, 嵌套的列表为子列表, 子列表的缺省外观和主列表的外

观是不一样的, 默认的形式是主列表是实心圆点, 第一级子类表使用空心的, 第二级子列表使用方框.

代码:

```
<html>
  <head><title>使用无序列表</title></head>
  <body>
    <h3>无序列表的样式</h3>
    <h5>默认样式</h5>
    <ul>
      <li>鼠标</li>
      <li>键盘</li>
      <li>主机</li>
      <li>显示器</li>
    </ul>
    <h5>定制样式</h5>
    <ul type="square">
      <li>个人 PC 机</li>
      <li>笔记本</li>
      <li type="circle">小型服务器</li>
      <li type="disc">掌上电脑</li>
    </ul>
    <h5>无序列表的嵌套</h5>
    <ul>
      <li>人体</li>
      <ul>
        <li>头</li>
        <li>手</li>
        <ul>
          <li>手指</li>
          <li>手掌</li>
        </ul>
        <li>腿</li>
      </ul>
      <li type="square">季节</li>
      <ul>
        <li type="circle">春</li>
        <li type="disc">夏</li>
        <li>秋</li>
        <li>冬</li>
      </ul>
    </ul>
```

```

        </ul>
    </body>
</html>

```

我想看一下:[htmlCode\使用无序列表.HTML](#)

地址 (D)  D:\备课资料\讲义\Web 基础_Html_C

无序列表的样式

默认样式

- 鼠标
- 键盘
- 主机
- 显示器

定制样式

- 个人PC机
- 笔记本
- 小型服务器
- 掌上电脑

无序列表的嵌套

- 人体
 - 头
 - 手
 - 手指
 - 手掌
 - 腿
- 季节
 - 春
 - 夏
 - 秋
 - 冬

5.8.2 使用有序列表

有序列表和无序列表类似,使用标签表示有序列表,嵌套使用表示列表项,指定type属性为有序的数字或字母,也可以对有序列表进行嵌套.

- type="1" 使用数字,缺省的情况
- type="a" 使用小写字母
- type="A" 使用大写字母
- type="i" 使用小写罗马字母
- type="I" 使用大写罗马字母
- type="X" 开始的编号

value ="" 可以改变当前值

代码:

```
<html>
  <head><title>使用无序列表</title></head>
  <body>
    <h3>有序列表的样式</h3>
    <h5>默认样式</h5>
    <ol>
      <li>鼠标</li>
      <li>键盘</li>
      <li value="10">主机</li>
      <li>显示器</li>
    </ol>
    <h5>使用字母</h5>
    <ol type="a">
      <li>个人 PC 机</li>
      <li>笔记本</li>
      <li>小型服务器</li>
      <li>掌上电脑</li>
    </ol>
    <h5>使用大写字母</h5>
    <ol type="A">
      <li>无序列表</li>
      <li>有序列表</li>
      <li>定义列表</li>
    </ol>
    <h5>使用小写罗马字母</h5>
    <ol type="i">
      <li>中国</li>
      <li>美国</li>
      <li>日本</li>
      <li>韩国</li>
    </ol>
    <h5>使用大写罗马字母</h5>
    <ol type="I">
      <li>萝卜</li>
      <li>白菜</li>
      <li>西红柿</li>
      <li>南瓜</li>
    </ol>
    <h5>设定开始值</h5>
    <ol start="5">
      <li>罗湖区</li>
      <li>宝安区</li>
```

```
<li>南山区</li>
<li>福田区</li>
<li>盐田区</li>
<li>龙岗区</li>
</ol>
<h5>有序列表和无序列表的嵌套</h5>
<ul>
  <li>人体</li>
    <ul>
      <li>头</li>
      <li>手</li>
        <ul>
          <li>手指</li>
          <li>手掌</li>
        </ul>
      <li>腿</li>
    </ul>
  <li type="square">季节</li>
    <ol type="I">
      <li>春</li>
      <li>夏</li>
      <li>秋</li>
      <li>冬</li>
    </ol>
</ul>
</body>
</html>
```

我想看一下:[htmlCode\使用有序列表.HTML](#)

有序列表的样式

默认样式

10

1. 鼠标
2. 键盘
10. 主机
11. 显示器

使用字母

- a. 个人PC机
- b. 笔记本
- c. 小型服务器
- d. 掌上电脑

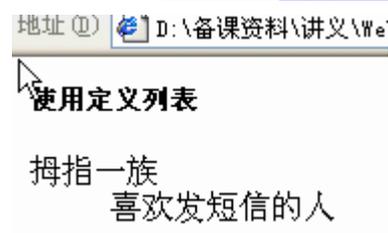
5.8.3 使用定义列表定义术语

使用<dl>列表标签可以定义术语,<dt>定义术语,<dd>定义的本身.

代码:

```
<html>
  <head><title>使用定义列表</title></head>
  <body>
    <h5>使用定义列表</h5>
    <dl>
      <dt>拇指一族</dt>
      <dd>喜欢发短信的人</dd>
    </dl>
  </body>
</html>
```

我想看一下:[htmlCode\定义列表.HTML](#)



5.9 使用链接

使用链接显示的文本,通过链接指向 Web 中的不同文件.

链接到其它网站:搜索

5.10 修饰文字

下标标签,显示下标,上

标标签,显示上标. 黑体

标签,使用黑体 斜体标签

<i></i>使用斜体 下划线标签

<u></u>使用下划线 插入标签

<ins></ins>标出插入文本 删除文本

代码:

```
<html>
  <head><title>修饰文本</title></head>
  <body>
    <b>使用黑体</b><br>
    <i>使用斜体</i><br>
```

水的分子式:H₂O

y=x³+5

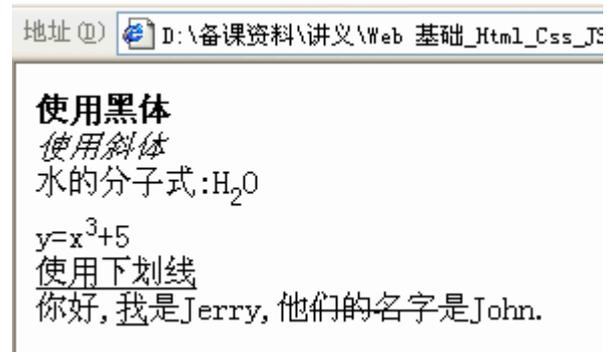
<u>使用下划线</u>

你好,<ins>我</ins>是 Jerry,他们的名字是 John.

</body>

</html>

我想看一下:[htmlCode\修饰文本.HTML](#)



第六讲 在网页中使用链接

自然界的蜘蛛用吐出的丝织出它赖以生存的网,我们这一讲将学会使用链接来编织我们奇妙的网络.这一讲我们会学到以下内容:

为链接添加辅助信息 利于链接在页面间跳转 指定窗口打开链接 为链接指定快捷键

6.1 为链接弹出辅助信息

使用 title 属性可以显示链接资源的说明信息,当鼠标移动链接上时会显示说明信息.

```
<a href = "http://www.bokee.com" title = "中国最大的博客网站">博客中国</a>
```

6.2 使用 Name 属性在文本内作标签

使用 name 属性可以帮助你快速定位的文档的各个部分,这对于 Html 的页面很长时,定位到用户感兴趣的内容很重要.

首先定义一个位置:

在页面的任何处使用 便会链接到 name 属性标识的位置.

6.3 使用 target 属性

指定在那个窗口打开链接,不指定默认在当前窗口打开链接,如果在新窗口打开链接使用在新窗口中打开链接,如果要在指定的窗口打开链接可以使用 target="窗口名".

6.4 为链接指定快捷键

使用 accesskey 属性可以使用 alt+ 快捷键,定位到指定的链接,例子:深圳热线

6.5 链接的综合例子

代码:

```
<html>
  <head><title>使用链接</title></head>
  <body>
    <a name="top">向你推荐网站</a><br>
    鼠标移动到链接上时会显示辅助信息:<br>
```

```

<a href="http://www.bokee.com" title="中国最大的博客网站">博客中国</a><br>
<p>
<b>网站介绍:</b><br>
<i>
该网站是博客们发表文章的窗口,有方兴东始创,这里聚集着大量的博客爱好者,<br>
内容广泛涉及到各个方面,很多知名的人士在上面都辟有专栏,<br> 一些观点,思想在这里碰撞.如果你想写下你的思想,你的观点,你的喜怒哀乐,<br> 你可以来到这里尽情挥洒,闲暇时你也可以到这里倾听陌生人的心声.
</i>
</p>
在当前窗口打开链接.<br>
<a href="http://www.octit.cn">北软教育门户网站</a><br>
<p>
<b>网站介绍:</b><br>
<i>
该网站是北软教育的门户网站,北软教育主要从事 It 培训教育,开发自己的教育产品,<br>
北软教育是深圳计算机行业协会的联办单位,以深圳为依托,向深圳的 It 企业输送人才,<br>
在就业问题是具有很大的优势.<br>
北软的教师也多来自深圳 It 界知名企业,具有丰富的软件研发经验,<br> 课程也主要结合项目开发,提供学员的动手能力,培养 It 企业急需的技能,<br> 学员具有很强的竞争能力.
</i>
</p>
在新窗口中打开链接.<br>
<a href="http://www.csdn.net" target="_blank">中国程序员网站</a><br>
<p>
<b>网站介绍:</b><br>
<i>
中国程序员的网络聚集地,无论你是刚刚试飞的,步入程序员大家庭的菜鸟,<br>
还是在羽翼丰满,在职场上打拼的老鸟,在这里都有你的伙伴,一起研究,一起争论,<br>
一起学习,一起成长,这里就是程序员的家.
</i>
</p>
在指定窗口打开链接.<br>
<a href="http://www.szonline.net" target="mywindows">深圳热线</a><br>
<p>
<b>网站介绍:</b><br>
<i>
深圳热线,发布深圳的新闻时事,娱乐信息,是深圳众多媒体中较有特色的网络

```

媒体,

国内,国际新闻时时更新,在这里可以扩充你的视野,是深圳人了望世界的窗

口.

</i>

</p>

为链接指定快捷键.

中国人才热线

<p>

网站介绍:

<i>

非常具有影响力的人力资源网站,庞大的简历库,大量的人才,和企业在这里寻找自己的目标,

在这里只要你有能力,这里就会给你一个机会去实现你心中的梦想.

企业也可以在这里找到为自己创造价值的人才.

</i>

</p>

返回顶部

</body>

</html>

我想看一下:[htmlCode\使用链接.HTML](#)

第七讲 创建有声有色的网页

我们会在这一讲讲述图片和多媒体的用法,包括:

设置图片尺寸,边框,位置

如何设置图文混排 在网

页中插入多媒体 在网页

中使用背景音乐

7.1 在网页中使用图片

使用标签可以在你的网页中包含图片,是你的网页丰富多彩,更具有表现力.

在网页中嵌入图片:使用 alt 属性,可以在浏览器不能显示图片时显示替代的文本信息,比如:.

7.1.1 图像的对齐方式

从对齐方式上区分图像有两种方式: 行间图像:在文本行中的图片,如果图片很大,文本行会很高,而且有一大片空白. 浮动图像:文本在图像边会换行,图像很大时,文字会围绕图片进行自动换行. 行间图片的对齐方式:

align="top" 图像和文本的顶部对齐.

align="middle" 图像和文本的中部对齐.

align="bottom",缺省值,图像和文本的底部对齐.

浮动图像的对齐方式:

align="left" 图片在文本的左边

align="right" 图片在文本的右边

使用<br clear="left">可以在图片的右侧产生空白.

<br clear="right">可以在图片的左侧产生空白.

<br clear="right">可以在图片的两侧产生空白.

7.1.2 使用 width,height 设置图片的尺寸

使用 width(像素)宽度,height(像素)高度可以设置图像的显示尺寸,设定图像的尺寸,浏览器会很容易的显示图片.显示速度会加快.

```

```

7.1.3 为图片设置边框

我们使用 border 属性为图片增加边框,border="0"是没有边框,我们没有必要设置,系统默认是没有边框的.我们上面设置图片的尺寸是图片的实际尺寸,是不包括边框的尺寸的.

设定图片的边框为 1 可以使用 border="1".

7.1.4 使用 hspace,vspace 调整图片的位置

hspace:在图片的左侧和右侧添加空白,使用的是像素值.

vspace:在图片的上下方向上添加空白,使用的是像素值.

```

```

7.2 使用图像作为链接

可以使用图片作为链接:

7.3 常用图片格式介绍

我们在这里介绍网页中常用的两种图片格式 Gif 和 Jpeg:

GIF 格式:

后缀为 .gif,最多支持 256 种颜色,采用压缩方式和 zip 相同,所有使用 zip 压缩 gif 格式的文件是没有用的.支持透明和动画效果.

Jpeg 格式:

后缀为 .jpg 或 .jpeg,是一种有损压缩格式,支持的颜色比 gif 丰富得多,但不支持透明效果和动画.

7.4 使用图片的综合例子

代码:

```
<html>
  <head><titl>在网页中使用图片</title></head>
  <body>
    <h4>在网页中使用图片</h4>
    <img src ="girl.jpg" alt="小女孩卡通图片"><br>
    <hr>
    行间图片的对齐方式<br>
    和文字的顶部对齐:<img src ="girl.jpg" alt="小女孩卡通图片" align="top"><br>
    和文字的中部对齐:<img src ="girl.jpg" alt="小女孩卡通图片"
align="middle"><br>
    和文字的底部对齐:<img src ="girl.jpg" alt="小女孩卡通图片" align="bottom">
    <hr>
    浮动图片的对齐方式<br>
    <img src ="girl.jpg" alt="小女孩卡通图片" align="left">左对齐:左对齐:左对齐:
左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:
```

左对齐:

左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:

使用 clear 属性

```
<br clear="left">
```

```
<hr>
```

```
左对齐:左对齐:左对齐:
```

```
左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:
```

左对齐:

左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:

没有使用 clear 属性

```
<hr>
```

在此处使用 clear 属性:

```
<br clear="left">
```

```

```

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

在此处使用 clear 属性

```
<br clear="right">
```

```
<hr>
```

设置图片的尺寸为 50X100:

```

```

```
<hr>
```

带边框的图片:

```
<hr>
```

调整图片的位置:


```

```

```
<hr>
```

使用图像作为链接:

```
<a href="http://www.google.com"></a>
```

```
</body>
```

```
</html>
```

我想看一下:[htmlCode\在网页中使用图片.HTML](#)

在网页中使用图片

在网页中使用图片



左对齐:左对齐:左对齐:左
左对齐:左对齐:左对齐:左对

在此处使用clear属性:

图片在右侧:图片在右侧:图片在右侧:图片在
右侧:图片在右侧:图片在右侧: 图片在右侧:图
图片在右侧: 图片在右侧:图片在右侧:图片在
处使用clear属性

7.5 使用链接嵌入多媒体

使用链接可以简单的嵌入多媒体文件:丁香花

使用背景音乐: <bgsound src=“dxh.wma” loop=3>只有 IE 支持.

代码:

```
<html>
  <head><title>在网页中嵌入多媒体</title></head>
  <body>
    使用背景音乐:<br>正在播放丁香花.
    <bgsound src=“dxh.wma” loop=3>
  </body>
</html>
```

我想看一下:[htmlCode\使用背景音乐.HTML](#)

嵌入多媒体对象:使用<object>标签,data 属性设定多媒体源,width,height 设定显示长高像素值,align 设定对齐方式支持 left(左对齐),right(右对齐),center(居中),hspace,vspace 指定多媒体显示画面的四周的空白间距,可以嵌套图片信息,当多媒体不能被识别时,显示图片信息的内容.

```
<object data=“dxh.wma” width=“200” height=“200” align=“left” hspace=“10” vspace=“20”>
  <img src=“girl.jpg” alt=“播放丁香花”>
</object>
```

代码:

```

<html>
  <head><title>在网页中嵌入多媒体</title></head>
  <body>
    使用链接<br>
    <a href="dxh.wma">丁香花</a><br>
    使用 object 对象嵌入多媒体<br>
    <object data="dxh.wma" width="200" leight="200" align="left" hspace="10"
vspace="20">
      
    </object>

  </body>
</html>

```

我想看一下:[htmlCode\在网页中嵌入多媒体\(object\).HTML](#)

使用 embed 嵌入多媒体:使用 embed 标签,src 属性指定多媒体源,autastart 设定是否自动播放,如设置成背景音乐可设置 autostart="true",loop 属性设定是否循环播放,volume 设定播放的音量, width,height 设定媒体的显示大小,入设置成背景音乐可以设置两者都为 0.

```
<embed src="dxh.wma" autostart="false" loop="true" vollume="10" width ="300" height="30">
```

代码:

```

<html>
  <head><title>在网页中嵌入多媒体</title></head>
  <body>
    使用链接<br>
    <a href="dxh.wma">丁香花</a><br>
    使用 embed 对象嵌入多媒体<br>
    <embed src="dxh.wma" autostart="false" loop="true" vollume="10" width ="300"
height="30">
  </body>
</html>

```

我想看一下:[htmlCode\在网页中嵌入多媒体\(embed\).HTML](#)

第八讲 用表格展示你的信息

用表格展示信息,具有很强的吸引力和灵活性,表格的使用在网页上已经相当的流行,我们在这一讲将主要关注以下内容:

表格的结构介绍 表格中数据的布局 单元格的合并 为表格设置背景 使用嵌套的表格

8.1 表格简介

表格是以行和列的方式来组织数据的,我们看一下下面的表格:

全国各大城市 8 月 8 日气温

城市	最高温度	最低温度
深圳	35	29
广州	36	28
北京	29	25
长春	27	19

在这张表中:

表的标题(可选的):全国各大城市 8 月 8 日气温 表的行:
表的水平方向上的数据 表的列:表的竖直方式上的数据
表的单元(cell):行和列的交点处 表的题头(表头):表的第一行如:城市,最高气温,最低气温.

8.2 使用标签表示 html 中的表

表格的一般标签和属性

<table>:创建表格
<caption>:表的标题(可选的),表格内容的描述信息
<tr>:表的行,包括表头和表的数据行
<td>:表的列
<th>:表头(可选的)会用黑体显示

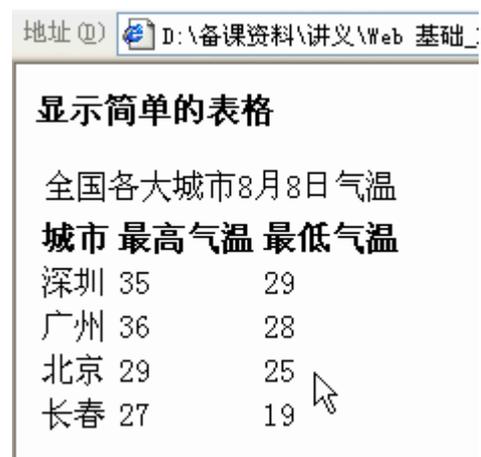
在 html 页面上显示全国各大城市 8 月 8 日气温:

代码:

```
<html>
  <head><title>使用表格</title></head>
  <h4>显示简单的表格</h4>
  <table>
    <!-- 这里是表的标题信息,表的标题信息是可选的 -->
    <caption>全国各大城市 8 月 8 日气温</caption>
```

```
<!-- 表的第一行,表头也是可选的 -->
<tr>
  <th>城市</th>
  <th>最高气温</th>
  <th>最低气温</th>
</tr>
<!-- 表的数据信息 -->
<tr>
  <td>深圳</td>
  <td>35</td>
  <td>29</td>
</tr>
<tr>
  <td>广州</td>
  <td>36</td>
  <td>28</td>
</tr>
<tr>
  <td>北京</td>
  <td>29</td>
  <td>25</td>
</tr>
<tr>
  <td>长春</td>
  <td>27</td>
  <td>19</td>
</tr>
</table>
</html>
```

我想看一下:[htmlCode\使用表格.HTML](#)



8.2.1 设定表格的边框

上面的表格是没有边框的,我们使用 border 属性可以为表格增加边框<table border>(默认宽度为 1)或设定边框的值<table border="10">,使用 border="0"相当于没有边框.

8.2.2 设定表格的对齐方式

使用 align 可以设定表格的对齐方式,缺省是左对齐的.

align="left" 左对齐

align="right" 右对齐

align="center" 居中

8.2.3 设定表格和单元格的宽度和高度

使用 width 可以设定表格的宽度,使用像素值 width="80",也可以使用百分比,如 width="100%"(屏幕宽度),width="50%"(屏幕一半的宽度),width="150%"(屏幕一倍半的宽度,会出现滚动条)

使用 height 可以设定表格的高度,height="150"

在<td><th>中使用 width 和 height 属性可以设置单元格的宽度和高度.

8.2.4 控制单元格距离

cellspacing:设置单元格间的间距

cellpadding:设置单元格内的间距

间距使用像素值.

8.2.5 表格中数据的布局

对于<tr><td><th>标签可以使用 align 确定文本的水平对齐方式:

align="left" 左对齐 (默认)

align="right" 右对齐

align="center" 水平居中

使用 valign 设定垂直方向上的对齐方式

valign="top" 顶端对齐

valign="middle" 垂直居中

valign="bottom" 底部对齐

valign = "baseline" 基线对齐(和顶端对齐方式类似,但要求当字体大小不一致时要保持底线是对齐的.

当单元格内的对齐方式和表格行的对齐方式有冲突时,单元个内的对齐方式优先.

8.2.6 为表格设置背景颜色

在<table><tr><td><th>中可以使用 bgcolor 设置表格或单元格的背景颜色

```
<table bgcolor="red">,<tr bgcolor="#00FF00">
```

8.2.7 禁止单元格中的文本折行

在<td>和<th>中使用 nowrap 可以是单元格内的文本在一行显示,不会折行显示.不要期望在<tr>中加入 nowrap 使一行单元个不折行,nowrap 对<tr>是不起作用的.

8.2.8 合并单元格

在<td>和<th>中使用 colspan 可以合并列,使用 rowspan 可以合并行.

```
<td colspan="2" rowspan="2"> 合并 2 列,2 行.
```

8.2.9 表格的嵌套

表格的嵌套是把一张表插入到另外一张表中.

```
<table>
  <tr>
    <td>
      <table>
        <tr>
          <td></td>
        </tr>
      </table>
    </td>
    <td>
    </td>
  </tr>
</table>
```

8.2.10 为表格和单元格加入背景

在<table>中使用 background 属性可以设置表格的背景图片,在<td>或<th>中使用 background 属性也可以为单元格设置背景图片.

8.2.11 使用在 width 中使用"*"

在 width 属性中我们可以指定像素值,使用百分比,在 html4.0 中我们也可以使用"*"来表示宽度,<col width="*">表示占满全部可用的空间,如果两列都使用"*,表示各占 50%宽度.使用"0*"表示占用最小的宽度,使用"1*"和"2*"等表示,如果有两列,一列用"1*",另外一列用"2*"表示第一列占 1/3,第二列占 2/3.

8.2.12 表格各个属性设置应用

我们前面已经讲述了表格的各个属性,我们可以看一下结合这些属性如何书写代码,显示表格的应用属性后的效果.

代码:

```
<html>
  <head><title>使用表格</title></head>
  <h4>显示简单的表格</h4>
  <table>
    <!-- 这里是表的标题信息,表的标题信息是可选的 -->
    <caption>全国各大城市 8 月 8 日气温</caption>
    <!-- 表的第一行,表头也是可选的 -->
    <tr>
      <th>城市</th>
      <th>最高气温</th>
      <th>最低气温</th>
    </tr>
    <!-- 表的数据信息 -->
    <tr>
      <td>深圳</td>
      <td>35</td>
      <td>29</td>
    </tr>
    <tr>
      <td>广州</td>
      <td>36</td>
      <td>28</td>
    </tr>
    <tr>
      <td>北京</td>
      <td>29</td>
      <td>25</td>
    </tr>
    <tr>
      <td>长春</td>
      <td>27</td>
```

```

        <td>19</td>
    </tr>
</table>
<h4>带边框的表格</h4>
<table border>
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr>
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr>
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
  </tr>
  <tr>
    <td>广州</td>
    <td>36</td>
    <td>28</td>
  </tr>
  <tr>
    <td>北京</td>
    <td>29</td>
    <td>25</td>
  </tr>
  <tr>
    <td>长春</td>
    <td>27</td>
    <td>19</td>
  </tr>
</table>
<h4>改变表格的边框</h4>
<table border="5">
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr>
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>

```

```
</tr>
<!-- 表的数据信息 -->
<tr>
  <td>深圳</td>
  <td>35</td>
  <td>29</td>
</tr>
<tr>
  <td>广州</td>
  <td>36</td>
  <td>28</td>
</tr>
<tr>
  <td>北京</td>
  <td>29</td>
  <td>25</td>
</tr>
<tr>
  <td>长春</td>
  <td>27</td>
  <td>19</td>
</tr>
</table>
<h4>表格的对齐方式</h4>
<table border align="left">
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr>
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr>
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
  </tr>
  <tr>
    <td>广州</td>
    <td>36</td>
    <td>28</td>
  </tr>
```

```
</table>
<table border align="right">
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr>
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr>
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
  </tr>
  <tr>
    <td>广州</td>
    <td>36</td>
    <td>28</td>
  </tr>
</table>
<table border align="center">
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr>
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr>
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
  </tr>
  <tr>
    <td>广州</td>
    <td>36</td>
    <td>28</td>
  </tr>
</table>
<h4>设定表格为 300 个像素</h4>
```

```
<table border width="300">
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr>
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr>
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
  </tr>
  <tr>
    <td>广州</td>
    <td>36</td>
    <td>28</td>
  </tr>
</table>
<h4>设定表格为屏幕宽度</h4>
<table border width="100%">
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr>
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr>
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
  </tr>
  <tr>
    <td>广州</td>
    <td>36</td>
    <td>28</td>
  </tr>
</table>
<h4>设定表格为 1.5 倍屏幕宽度</h4>
```

```
<table border width="150%">
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr>
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr>
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
  </tr>
  <tr>
    <td>广州</td>
    <td>36</td>
    <td>28</td>
  </tr>
</table>
```

<h4>设定单元格间的间距</h4>

```
<table border="2" cellspacing="10">
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr>
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr>
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
  </tr>
  <tr>
    <td>广州</td>
    <td>36</td>
    <td>28</td>
  </tr>
</table>
```

<h4>设定单元格内的间距</h4>

```

<table border="2" cellpadding="10">
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr>
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr>
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
  </tr>
  <tr>
    <td>广州</td>
    <td>36</td>
    <td>28</td>
  </tr>
</table>

```

<h4>设定单元格内文本对齐</h4>

```

<table border width="100%" height="150">
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr align="right" valign="top">
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr align="center" valign="middle">
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
  </tr>
  <tr align="left" valign="bottom">
    <td align="right" valign="middle">广州</td>
    <td align="left" valign="baseline">36</td>
    <td>28</td>
  </tr>
</table>

```

<h4>设定表格的背景</h4>

```

<table border width="100%" bgcolor="blue">
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr bgcolor="green">
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr >
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
  </tr>
  <tr >
    <td>广州</td>
    <td bgcolor="red">36</td>
    <td>28</td>
  </tr>
</table>

```

<h4>单元格内的文本禁止折行</h4>

```

<table border width="30%" >
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr >
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr >
    <td>深圳</td>
    <td nowrap>35.677777888888888888888888</td>
    <td>29</td>
  </tr>
  <tr >
    <td>广州</td>
    <td>36</td>
    <td>28</td>
  </tr>
</table>
<h4>合并单元格</h4>

```

```
<table border width="50%">
<!-- 这里是表的标题信息,表的标题信息是可选的 -->
<caption>全国各大城市 8 月 8 日气温</caption>
<!-- 表的第一行,表头也是可选的 -->
<tr>
  <th colspan="2">城市</th>
  <th>最低气温</th>
</tr>
<!-- 表的数据信息 -->
<tr >
  <td rowspan="2">深圳</td>
  <td>35</td>
  <td>29</td>
</tr>
<tr>
  <td>36</td>
  <td>28</td>
</tr>
<tr>
  <td>北京</td>
  <td>29</td>
  <td>25</td>
</tr>
<tr>
  <td>长春</td>
  <td>27</td>
  <td>19</td>
</tr>
</table>
<h4>表格的嵌套</h4>
<table border width="30%">
  <tr>
    <td>
      <table border bgcolor="blue">
        <tr>
          <td>表二</td>
          <td>表二</td>
          <td>表二</td>
          <td>表二</td>
        </tr>
        <tr>
          <td>表二</td>
          <td>表二</td>
          <td>表二</td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```

```

        <td>表二</td>
    </tr>
</table>
</td>
<td>表一</td>
<td>表一</td>
</tr>
<tr>
    <td>表一</td>
    <td>表一</td>
    <td>表一</td>
</tr>
<tr>
    <td>表一</td>
    <td>表一</td>
    <td>表一</td>
</tr>
</table>
<h4>带背景的表格</h4>
<table border background="caomei.jpg" width="50%">
<!-- 这里是表的标题信息,表的标题信息是可选的 -->
<caption>全国各大城市 8 月 8 日气温</caption>
<!-- 表的第一行,表头也是可选的 -->
<tr>
    <th>城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
</tr>
<!-- 表的数据信息 -->
<tr>
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
</tr>
<tr>
    <td>广州</td>
    <td>36</td>
    <td>28</td>
</tr>
<tr>
    <td>北京</td>
    <td>29</td>
    <td>25</td>
</tr>

```

```
<tr>
  <td>长春</td>
  <td>27</td>
  <td>19</td>
</tr>
</table>
<h4>单元格带背景的表格</h4>
<table border width="50%">
  <!-- 这里是表的标题信息,表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行,表头也是可选的 -->
  <tr>
    <th background="caomei.jpg">城市</th>
    <th>最高气温</th>
    <th>最低气温</th>
  </tr>
  <!-- 表的数据信息 -->
  <tr>
    <td>深圳</td>
    <td>35</td>
    <td>29</td>
  </tr>
  <tr>
    <td>广州</td>
    <td background="caomei.jpg">36</td>
    <td>28</td>
  </tr>
  <tr>
    <td>北京</td>
    <td>29</td>
    <td>25</td>
  </tr>
  <tr>
    <td>长春</td>
    <td>27</td>
    <td>19</td>
  </tr>
</table>
</html>
我想看一下:htmlCode\使用表格.HTML
```

第九讲 使用框架划分窗口

我们在前面讲述的是在整个窗口中显示一篇文档,我们在这一讲将使用框架技术,把浏览器的当前窗口划分几个区域,在每个区域显示不同的文档内容,我们在这一讲将讲述:

- 框架的用途 创建简单的框架
- 利用框架分割窗口 框架的嵌套 修改框架的显示属性
- 在框架中使用链接
- 使用预定义的框架名 浏览器不支持框架的处理

9.1 框架简介

利用框架可以在浏览器的主窗口中分割出多个子窗口显示不同的文档信息.框架可以水平分割,也可以垂直分割,每个框架默认可以带有滚动条,可以缩放尺寸.

9.2 创建 frameset 文档

frameset 文档和普通的 html 文档稍有不同,普通的 html 文档包含<head>和<body>元素,而 frameset 文档包含<head><frameset>元素.

我们可以建立一个简单的框架页面:

代码:

```
<html>
  <head><title>最简单的框架</title></head>
  <frameset cols="50%,50%">
    <frame src="first.html">
    <frame src="second.html">
  </frameset>
  <body>
    你的浏览器不支持框架!<br>
    <a href="first.html">文档 1</a><br>
    <a href="second.html">文档 2</a>
  </body>
</frameset>
</html>
first.html:
<html>
<head><title>first html</title></head>
<body>第一个 Html 文档</body>
```

```

</html>
second.html
<html>
<head><title>second html</title></head>
<body><b><i>第二个 Html 文档</i></b></body>
</html>

```

我想看一下:[htmlCode\最简单的框架.HTML](#)



9.3 <frameset>的用法

<frameset>包含一个或多个<frame>标签,frameset 使用 rows 和 cols 来水平和垂直分割页面.

<frame>标签使用 src 属性指定在此框架中要显示的文档.如果浏览器不支持框架会显示 <noframes>标签中的内容.

9.4 使用 cols 属性垂直分割窗口

<frameset>标签中的 cols 属性可以垂直分割窗口,定义两个或多个垂直框架的尺寸,各框架的尺寸值用“,”分隔.cols 的缺省值是 100%,若没有指定任何值,就会只有一个框架占据整个水平窗口.

每个框架的宽度可用是绝对宽度(使用像素或百分比),也可以是相对宽度(使用数字和“*”,如 2*).分隔原则是绝对值具有较高的优先级,最后按相对宽度划分剩余的空间.

```
<frameset cols="25%,75%">
```

```
<frameset cols="1*,3*">
```

<frameset cols="100,25%,2*,3*"> 第一个占用 100 个像素,第二个占用整个的 25%,第三个占剩下的 1/4,第四个占剩下的 3/4.

9.5 使用 rows 属性水平分割窗口

rows 的用法和 cols 类似,它进行水平分割窗口,也可以指定绝对和相对高度,这里的高度为整个窗口的高度

9.6 同时应用 cols 和 rows

在<frameset>中指定 cols 和 rows 会把页面划分成网格状.比如<frameset cols="200,*" rows="25%,25%,50%">将把窗口划分成 2X3=6 的网格.

代码:

```
<html>
  <head><title>最简单的框架</title></head>
  <frameset cols="200,*" rows="25%,25%,50%">
    <frame src="first.html">
    <frame src="second.html">
    <frame src="three.html">
    <frame src="forth.html">
    <frame src="fifth.html">
    <frame src="sixth.html">
  <noframes>
    <body>
      你的浏览器不支持框架!<br>
      <a href="first.html">文档 1</a> <br>
      <a href="second.html">文档 2</a> <br>
      <a href="three.html">文档 3</a> <br>
      <a href="forth.html">文档 4</a> <br>
      <a href="fifth.html">文档 5</a> <br>
      <a href="sixth.html">文档 6</a> <br>
    </body>
  </noframes>
</frameset>
</html>
three.html
<html>
<head><title>three html</title></head>
<body><i>第三个 Html 文档</i></body>
</html>
forth.html
<html>
<head><title>forth html</title></head>
<body><b><i>第四个 Html 文档</i></b></body>
</html>
fifth.html
<html>
<head><title>fifth html</title></head>
<body><font color="blue">第五个 Html 文档</font></body>
</html>
sixth.html
<html>
<head><title>sixth html</title></head>
<body><font color="green">第六个 Html 文档</font></body>
</html>
我想看一下:htmlCode\six框架.HTML
```



9.7 frameset 的嵌套

框架是可用嵌套的,被分割的框子叫父框架,包含的框架为子框架.下面的例子是把一页划分为 2 个框架,下面的框架有划分为两个框架.

代码:

```
<html>
  <head><title>框架的嵌套</title></head>
  <frameset rows="25%,75%">
    <frame src="first.html">
      <frameset cols="50%,50%">
        <frame src="second.html">
          <frame src="three.html">
        </frameset>
      </frameset>
    </frameset>
  </html>
```

我想看一下:[htmlCode\框架的嵌套.HTML](#)

9.8 设置边框颜色

在 frameset 和 frame 中使用 bordercolor 属性可以设定边框的颜色,缺省的颜色是灰色.<frameset bordercolor="#FF0000">,<frame bordercolor="#00FF00">

代码:

```

<html>
  <head><title>带边框颜色的框架</title></head>
  <frameset rows="25%,75%" bordercolor="blue" border="8">
    <frame src="first.html">
    <frameset cols="50%,50%">
      <frame src="second.html" bordercolor="green">
      <frame src="three.html">
    </frameset>
  </frameset>
</html>

```

我想看一下:[htmlCode\带边框颜色的框架.HTML](#)

第一个Html文档

第二个Html文档

第三个Html文档

9.9 在 frame 中插入文档

在 frame 标签中使用 src 属性插入要在该框架中显示的文档.

```
<frame src="插入的页面.html">
```

9.9.1 使用 frameborder 设置是否显示边框

在<frame>标签中设置 frameborder 属性可以决定是否显示框架与相邻的框架之间的边框是否显示.默认值为 1,显示,设为 0 表示不显示.如果要设置边框的尺寸可以在<frameset>中的 border 属性中设置.

设定框架的宽度:

代码:

```

<html>
  <head><title>带边框颜色的框架</title></head>
  <frameset rows="25%,75%" border="8">
    <frame src="first.html">
    <frameset cols="50%,50%">
      <frame src="second.html" >
      <frame src="three.html">
    </frameset>
  </frameset>
</html>

```

</html> 我想看一下:[htmlCode\设定边框的宽度.HTML](#)

不显示框架的情况:

代码:

```
<html>
  <head><title>不显示边框的框架</title></head>
  <frameset rows="25%,75%" >
    <frame frameborder="0" src="first.html">
    <frameset cols="50%,50%">
      <frame src="second.html" frameborder="0">
      <frame src="three.html" frameborder="0">
    </frameset>
  </frameset>
</html>
```

我想看一下:[htmlCode\不显示边框的框架.HTML](#)

9.9.2 marginheight,marginwidth 设置页面在框架中的位置

在<frame>标签中使用 marginheight,和 marginwidth 属性可以设定显示的页面距离框架的上下距离和左右距离.<frame src="first.html" marginheight="100" marginwidth="200">

代码:

```
<html>
  <head><title>调整文本在框架中的位置</title></head>
  <frameset rows="25%,75%" >
    <frame src="跳蚤和爬蚤的故事.HTML" marginheight="100" marginwidth="50">
    <frameset cols="50%,50%">
      <frame src="snow.HTML" marginheight="100" marginwidth="50">
      <frame src="爬楼和人生.HTML" marginheight="100" marginwidth="50">
    </frameset>
  </frameset>
</html>
```

我想看一下:[htmlCode\调整文本在框架中的位置.HTML](#)

9.9.3 控制滚动条,取消框架尺寸大小调整

禁止调整框架尺寸大小,默认的情况下,框架是可以调整大小的,在<frame>中使用 noresize 属性可以禁止调整框架尺寸大小.

```
<frame src="first.html" noresize>
```

控制滚动条,当框架中不能容纳要显示的内容时,框架中会出现水平,竖直的滚动条,在默认的情况下浏览器根据需要决定是否出现滚动条.我们也可以在<frame>标签中控制滚动条:

scrolling="yes" 总是出现滚动条

scrolling="no" 不要出现滚动条

scrolling="auto" 根据需要出现滚动条,默认值

代码:

```
<html>
  <head><title>noresize 的使用及控制滚动条</title></head>
  <frameset rows="25%,75%" >
    <frame src="跳蚤和爬蚤的故事.HTML" scrolling="yes">
    <frameset cols="50%,50%">
      <frame src="snow.HTML" noresize>
      <frame src="爬楼和人生.HTML" scrolling="no">
    </frameset>
  </frameset>
</html>
```

我想看一下:[htmlCode\框架滚动条和Noresize使用.HTML](#)

9.9.4 name 属性

使用 name 属性来命名框架,在其它框架中的链接可以使用这个名字来指向这个框架.

代码:

```
<html>
  <head><title>使用 name 属性</title></head>
  <frameset rows="25%,75%" >
    <frame src="跳蚤和爬蚤的故事.HTML" name="main">
    <frame src="snow.html">
  </frameset>
</html>
```

9.10 在框架中使用链接

我们既然为框架指定了 name 属性,那么我们就可以在链接的使用 target 属性来指向我们命名的框架.比如,我们可以建立一个分两个框架的页面,左面的页面是导航链接,右面的框架命名为"main"用于具体内容的显示.左边的链接需要指向右面的框架.当我们点击左面框架中的链接时,右面的框架不在显示初始的页面,而是显示对应的内容.如果没有找到匹配的框架名称时会新开窗口来显示链接的内容.

代码:

welcome.html

```
<html>
  <head><title>欢迎页面</title></head>
  <body>
    <center><b>欢迎光临北软教育</b></center><br>
    <center>深圳市北软技术有限公司</center>
    <center>地址:深圳市华侨城培训中心 3 楼</center>
    <center>兰州教育中心地址:兰州市西北书城 21 楼</center>
    <center>主营业务:IT 培训</center>
  </body>
```

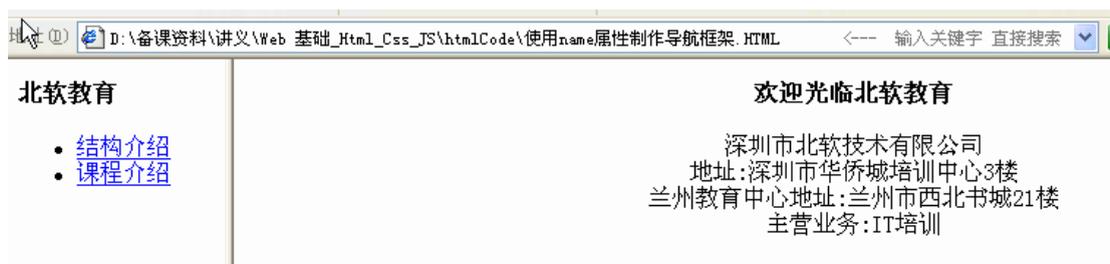


```

<html>
  <head><title>使用 name 属性制作导航框架</title></head>
  <frameset cols="15%,85%" >
    <frame src="navbar.HTML" >
    <frame src="welcome.html" name="main">
  </frameset>
</html>

```

我想看一下:[htmlCode\使用name属性制作导航框架.HTML](#)



9.10.1 使用<base>标签设定缺省的目标框架

我们经常在一个页面中的多个链接指向同一个框架,我们每个都使用 target="框架名"显然不是很方便.我们可以在文档的<head>标签中嵌套<base target="框架名">,使页面中的链接都指定到目标框架中.

我们可以修改上面的 navbar.html 就可以达到这个目的.

代码:

```

<html>
  <head>
    <title>导航条</title>
    <base target="main">
  </head>
  <body>
    <b>北软教育</b><br>
    <ul>
      <li><a href="introduce.html" >机构介绍</a></li>
      <li><a href="classIntroduce.html" >课程介绍</a></li>
    </ul>
  </body>
</html>

```

这样机构介绍和课程介绍的链接都会在 main 框架中显示,不用在为它们分别指定目标框架.

9.10.2 预定义的目标框架名

在 html 中预定义了四个目标框架名,它们都以下划线开头,并且区分大小写.

_blank:当 target="_blank"时,浏览器会新开窗口显示链接指定的信息.

```
<a href="first.html" target="_blank">new windows</a>
```

单击 new windows 链接时会在新窗口中显示内容.

_parent:在父框架中显示,如果没有父框架会在当前框架中显示信息.

```
<a href="first.html" target="_parent">display at parent windows</a>
```

_self:在当前框架中显示,它是配合<base target="框架名">使用,比如我们在<head>部分定义了<base target="main">,我们想让其中某个链接在当前窗口而不是 main 框架中打开,我们可以使用: current windows

代码:

```
<html>
  <head>
    <title>导航条</title>
    <b>base target="main"</b>
  </head>
  <body>
    <b>北软教育</b><br>
    <ul>
      <li><a href="introduce.html" >机构介绍</a></li>
      <li><a href="classIntroduce.html" >课程介绍</a></li>
      <li><a href="classIntroduce.html" target="_self">当前窗口显示</a></li>
    </ul>
  </body>
</html>
```

_top:去掉所有框架,全屏显示文档内容,如果要回到以前的框架结构需要安浏览器的后退按钮.

代码:

预定义框架名的应用.html

```
<html>
  <head><title>预定义框架名的应用</title></head>
  <frameset cols="20%,80%" >
    <frame src="first.HTML" >
    <frameset rows="2*,1*">
      <frame src="second.html">
      <frame src="docnavbar.html" >
    </frameset>
  </frameset>
</html>
```

docnavbar.html

```
<html>
  <head><title>导航条</title>
  <b>base target="_top"</b>
</head>
<body>
  <b>选择你喜欢的文章</b><br>
  <ul>
    <li><a href="跳蚤和爬蚤的故事.html" target="_self">跳蚤和爬蚤的故事</a></li>
```

```

    <li><a href="爬楼和人生.html" target="_blank">爬楼和人生</a></li>
    <li><a href="snow.html" target="_parent">江雪</a></li>
    <li><a href="热爱生命.html" >热爱生命</a></li>
  </ul>
</body>
</html>

```

热爱生命.html

```

<html>
  <head><title>热爱生命</title></head>
  <body>
    <b>热爱生命</b>
    <p> 我不去想是否能够成功
    <br> 既然选择了远方<br>
    便只顾风雨兼程
    </p>
    <p> 我不去想能否赢得爱情
    <br> 既然钟情于玫瑰<br>
    就勇敢地吐露真诚
    </p><p> 我不去想身后会不会袭来寒风
    冷雨<br> 既然目标是地平线<br> 留给
    世界的只能是背影
    </p><p> 我不去想未来是平坦还是
    泥泞<br> 只要热爱生命<br> 一
    切，都在意料之中
    </p></p>
    (汪国真)
  </p>

  </body>
</html>

```

我想看一下:[htmlCode\预定义框架名的使用.HTML](#)

9.11 <noframes>为不支持框架的浏览器提供替代内容

<noframes>标签为不支持框架的浏览器提供了替代的内容,当浏览器不支持框架时会显示<noframes>标签内的内容:

代码:

```

<html>
  <head><title>最简单的框架</title></head>
  <frameset cols="50%,50%">

```

```
<frame src="first.html">
<frame src="second.html">
<noframes>
  <body>
    你的浏览器不支持框架!<br>
    <a href="first.html">文档 1</a>
    <a href="second.html">文档 2</a>
  </body>
</noframes>
</frameset>
</html>
```

我们上面的代码会在浏览器不支持框架时显示两个文档的链接.

第十讲 利用表单和用户交互

利用表单和用户交互是 web 基础中最重要的内容,通过这一讲你将学会如何从用户那里获得你感兴趣的数据信息,我们在这一讲主要向你讲述:

如何建立你自己的表单 学会使用**<Input>**
标签建立各种表单控件 使用下拉列表和
多行文本框 定制你的按钮

10.1 表单介绍

web 应用程序经常要收集用户的信息,如登陆名,密码,Email 信息等,在 Html 中使用表单来收集这些信息.

表单是信息域的集合,这些信息域有着不同的表现形式,比如,文本框,单选按钮,下拉框等等.

10.2 建立一个最简单的表单

我们建立一个最简单的表单,在网页上输入你的学号,并把学号发送给指定的邮箱(lxz003@126.com)

代码:

```
<html>
<head><title>最简单的表单</title></head>
<body>
  <h4>收集学员的信息</h4>
  <form action="mailto:lxz003@126.com" method="post">
    请输入你的学号:
    <input type="text" value="" size="10" name="studentID"><br>
    <input type="submit" value="发送"><input type="reset" value="重填">
  </form>
</body>
</html>
```

我想看一下:[htmlCode\最简单的表单.HTML](#)

地址 (D)  D:\备课资料\讲义\Web_基础_Html

收集学员的信息

请输入你的学号:

在这段代码中<form>是表单的标签,action 属性指定处理表单的 url,method 指定发送表单使用的方法,<input type="text">是文本输入框,<input type="submit"> 是提交按钮,<input type="reset">是重置按钮.重置按钮会清除刚刚填入的内容,提交按钮会提交填入的内容给指定的 url 去处理表单.我们这里是把用户填入的学号信息以邮件的形式发送到指定地点.

10.3 <form>标签

<form>表单标签,一个页面中可以含有多个表单标签,被<form>包含的是一组要输入和提交给表单程序处理的信息.

method 属性:指定表单提交的方法,使用两种方法:

get 方法:默认的方法,把用户提交的数据附加在 Url 地址后面以查询串的方式发送数据.

post 方法:用户输入的信息以数据包的方式发送数据

使用 get 方法和 post 方法的区别:

使用 get 方法数据信息在地址栏中显示,不够安全,发送的字节数也有限制.

post 方法以数据包的方式发送较 get 安全些,发送的数据量大些,但传送的速度较 get 要慢.

10.4 表单控件标签的用法

<input>创建文本框或按钮

<textarea>创建多行文本区

<select>创建下拉框

<button>创建按钮

10.4.1 <input>标签的使用

<input>标签中的 type 属性指定表单控件的类型,name 属性用来标识当前的表单控件,通过 name 引用表单控件,获得表单控件的值.没有 name 属性将无法提交任何值.

type 类型共有 10 种不同的类型:

text:创建单行文本框 用 value 指定默认值,size 指定文本框的大小,maxlength 指定文本框最多能输入的字符数.

password:密码框和 text 类似,但输入内容不直接显示在屏幕上用"*"屏蔽.

checkbox:复选框,可以同时选择多项.

radio:单选框,一次只能选择一项 submit:提交

按钮,提交表单 image:图片形式的提交按钮

reset:重置按钮,把用户输入的信息置为初始值

button:创建一个按钮,但不能自动提交

hidden:隐藏域,不显示在屏幕上,可以用来收集不需要用户看到的信息

file:文件上传,发送客户端的文件

10.4.1.1 单行文本框的用法

```
<input type="text" name="" value="" size="" maxlength="">
```

value:为文本框设定初始值. size:指定文本框显示的长度,按英文字符个数为单位. maxlength:允许输入的最大英文字符数

我们可以使用 readonly 把单行文本框变成只读不允许用户改变初始值,使用 disable 可以把文本框变灰,处于不可用状态.

10.4.1.2 密码输入框

```
<input type="password" name="" maxlength="">
```

密码输入框和文本框类似,但输入的信息在屏幕上显示的是替代信息,防止别人看到输入的内容,但是实际信息也是未加密的.

10.4.1.3 复选框

`<input type="checkbox" name="" value="">` 要使复选框在初始状态下选中,使用 `checked` 属性:`<input type="checkbox" name="" value="" checked>` 只有被选中的复选框在提交时,其数据才被提交.

10.4.1.4 单选按钮

`<input type="radio" name="" value="">` 表示单选按钮,在一组单选按钮中只能选中一个按钮,如果设置按钮默认是选中状态使用 `checked` 属性,和复选按钮一致.

10.4.1.5 提交按钮

`<input type="submit" value="" name="">` `value` 属性是按钮上的显示值,提交按钮的作用是提交表单数据,提交表单可以放在表单的任何位置,而且一张表单可以有多个提交按钮,使用 `size` 属性可以控制按钮的尺寸.

10.4.1.6 图形提交按钮

`<input type="image" src="XX.gif" alt="" align="">` 图形提交按钮的功能和提交按钮的功能类似,以图片代替灰色的按钮,属性和我们前面讲的 `img` 的具有的属性类似.`src` 指定使用的图片,`alt` 显示提示文本,`align` 指定对齐方式.

10.4.1.7 重置按钮

`<input type="reset" value="" name="" size="">` 重置按钮的作用是把表单中的数据置为初始值,用户填入的数据会放弃,我们在使用重置按钮时,尽量不要把重置按钮放在提交按钮前,以免用户不小心按了重置按钮丢失已经填入的数据.

10.4.1.8 隐藏域

`<input type="hidden" name="" value="">` 隐藏域和单行文本框差不多,但是隐藏域的信息是不显示给用户的,用户没有机会改动隐藏域的信息,隐藏域是为了收集程序需要的信息,在提交时一起提交给程序处理.

10.4.1.9 按钮

`<input type="button" value="" >` 创建一个按钮,没有缺省的行为事件,当使用 `onclick="XX()"` 时会在点击时执行 XX 事件,我们在脚本中去实现 XX()方法.

10.4.1.10 文件上传对话框

`<input type="file" name="">`显示文件上传对话框,提交数据时会一起提交.

10.4.2 input 标签的综合例子

下面是综合使用 input 标签的例子:

代码:

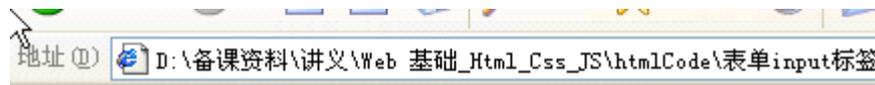
```
<html>
<head><title>input 标签的使用</title></head>
<body>
  <h4>用户注册</h4>
  <form action="mailto:lxz003@126.com" method="post" name="usrform">
  <table width="50%">
    <tr>
      <td width="20%">用户名:</td>
      <td><input type="text" name="username" value="" size="30"
maxlength="30"></td>
    </tr>
    <tr>
      <td width="20%">密码:</td>
      <td><input type="password" name="userpassword" size="30"
maxlength="30"></td>
    </tr>
    <tr>
      <td width="20%">性别:</td>
      <td><input type="radio" name="sex" value="0" checked>男 &nbsp;&nbsp;
      <input type="radio" name="sex" value="1" >女</td>
    </tr>
    <tr>
      <td width="20%">爱好:</td>
      <td><input type="checkbox" name="instresting" value="readBook"> 读书
&nbsp;&nbsp;&nbsp;
      <input type="checkbox" name="instresting" value="web"> 上网&nbsp;&nbsp;&nbsp;
      <input type="checkbox" name="instresting" value="music"> 听音乐
&nbsp;&nbsp;&nbsp;
    </td>
  </tr>
  </table>
</form>
</body>
</html>
```

```

</tr>
<tr>
  <td width="20%">靓照:</td>
  <td><input type="file" name="photo" size="30"></td>
</tr>
<tr>
  <td width="20%"></td>
  <td></td>
</tr>
<tr>
  <td colspan="2"><input type="submit" name="register" value="我填好了">
  <input type="button" name="checkuser" value="检测我的用户名">
  <input type="reset" value="重填">
  <input type="image" src="backButton.bmp" width="80" height="30"
name="goback" alt="返回上页">
  </td>
</tr>
</table>
<input type="hidden" name="option" value="register">
<input type="hidden" name="checkNumber" value="0121">
</form>
</body>
</html>

```

我想看一下:[htmlCode\表单input标签的使用.HTML](#)



用户注册

用户名:

密码:

性别: 男 女

爱好: 读书 上网 听音乐

靓照:



10.4.3 下拉框

使用<select>标签可以创建下拉框,<select>标签至少要嵌套一个<option>选项标签,选项标签会在下拉框中创建一个供选择的条目,在<select>标签中指定 size 属性下拉框便成为列表

框,size 的值就是列表框显示的条目,其它条目要通过拉动滚动条才可以看到或选取.当 size 大于条目的总数目时,会在用空的条目补充.缺省情况下,下拉框和列表框只可以选择一项条目,指定 multiple 后可以选择多个条目.

<select>标签要指定 name 属性,option 可以指定 value 属性,使用 selected 表示默认被选择.也可以使用 disable 设定是不可选择的,使用 tabindex 改变列表的顺序.

使用格式:

```
<select name="" size="" multiple>
  <option value="" selected>XXX
  <option value="" >YYY
</select>
```

10.4.4 多行文本框

<textarea>多行文本框,允许用户输入多行文本,input 的 text 类型只能输入一行文本.使用 rows 指定多行文本框显示的行数,cols 指定多行文本框显示的列数,当输入的字符超出指定的行数和列数时会出现滚动条.

```
<textarea rows="10" cols="50" name="" >默认文本</textarea>
```

10.4.5 使用 button 创建按钮

<button>的 type 属性可以是 button(普通按钮),reset(重置按钮),submit(提交按钮),和 input 标签中的按钮类似,但<button></button>中可以嵌套图片,表格,文本等作为按钮内容,显示为一个按钮.

10.4.6 readonly 和 disable 属性

我们可以给不希望用户修改的表单控件增加 readonly 属性,对于不允许用户操作的表单控件增加 disable 属性.比如我们可以把文本框,密码框设置为 readonly 不允许用户更改,可也把某些按钮设置成 disable 不允许用户点击.

10.4.7 设置表单顺序和快捷键

使用 tabindex 属性可以设定 tab 键的顺序,使用该属性可以使表单控件按顺序响应,使用 accesskey 属性可以为表单控件指定快捷键,在 windows 使用 alt+快捷键使特定的控件获得焦点.

10.5 表单的处理

我们前面讲述的表单例子都是使用发邮件的形式处理数据的,实际上我们可以使用客户端脚本语言处理表单数据,比如我们可以使用 javascript 语言对用户输入的数据进行可靠性

和完整性的验证等.表单的处理方式我们会在 JavaScript 语言以及后续的 jsp 课程中进行介绍.

10.6 表单综合例子

下面的例子是一个 Java 学员调查表的例子,我们将综合应用各种表单控件.用户填入的信息会被发送给指定的邮箱,用于分析数据:

代码:

```
<html>
  <head><title>Java 学员调查表</title></head>
  <body>
    <h4>Java 学员调查表</h4>
    <form action="mailto:lxz003@126.com" method="post" name="usrform">
      <table width="50%">
        <tr>
          <td width="20%">姓名:</td>
          <td><input type="text" name="username" value="" size="30"
maxlength="30"></td>
        </tr>
        <tr>
          <td width="20%">性别:</td>
          <td><input type="radio" name="sex" value="0" checked>男&nbsp;&nbsp;&nbsp;
            <input type="radio" name="sex" value="1" >女</td>
        </tr>
        <tr>
          <td width="20%">地址:</td>
          <td><input type="text" name="address" value="" size="30"
maxlength="50"></td>
        </tr>
        <tr>
          <td width="20%">电话:</td>
          <td><input type="text" name="tel" value="" size="30"
maxlength="11"></td>
        </tr>
        <tr>
          <td width="20%">通过什么渠道认识我们:</td>
          <td><input type="checkbox" name="whercome" value="lznews">兰州商报
&nbsp;&nbsp;&nbsp;
            <input type="checkbox" name="whercome" value="web">网络&nbsp;&nbsp;&nbsp;
            <input type="checkbox" name="whercome" value="TV">电视&nbsp;&nbsp;&nbsp;
            <input type="checkbox" name="whercome" value="sheet">宣传单
&nbsp;&nbsp;&nbsp;<br>
            <input type="checkbox" name="whercome" value="friend">朋友介绍
&nbsp;&nbsp;&nbsp;
            <input type="checkbox" name="whercome" value="other"> 其他
```

```

&nbsp;
        </td>
    </tr>
    <tr>
        <td width="20%">目前薪资:</td>
        <td>
            <select name="salary">
                <option value="0" >学生</option>
                <option value="1" selected>1000 以下</option>
                <option value="2" >1000-2000</option>
                <option value="3" >2000-5000</option>
                <option value="4" >5000 以上</option>
            </select>
        </td>
    </tr>
    <tr>
        <td width="20%">你学过那些课程:</td>
        <td>
            <select name="class" size="6" multiple>
                <option value="0" selected>J2se</option>
                <option value="1" >Html</option>
                <option value="2" selected>Javascript</option>
                <option value="3" >Jsp&Servlet</option>
                <option value="4" >Oracle</option>
                <option value="5" >DB2</option>
                <option value="6" >Struts</option>
                <option value="7" >J2me</option>
            </select>
        </td>
    </tr>
    <tr>
        <td width="20%">你对中心的建议:</td>
        <td>
            <textarea name="yourword" rows="5" cols="40">我们希望:</textarea>
        </td>
    </tr>
    <tr>
        <td colspan="2"><input type="submit" name="register" value="我填好了"
">&nbsp;
        <button name="checkuser" ></button>
        <input type="reset" value="重填">
    </td>
</tr>

```

```

</table>
<input type="hidden" name="option" value="register">
<input type="hidden" name="checkNumber" value="0121">
</form>
</body>
</html>

```

我想看一下:<htmlCode\Java学员调查表.HTML>

Java学员调查表

姓名:

性别: 男 女

地址:

电话:

通过什么渠道认识我们: 兰州商报 网络 电视 宣传单
朋友介绍 其他

目前薪资: 1000以下

你学过那些课程:
 J2se
 Html
 Javascript
 Jsp&Servlet
 Oracle
 DB2

你对中心的建议: 我们希望:

到这里我们的 Html 基础课程就结束了,运用上面的知识我们可以构造自己的 Html 页面.通过自己的网页去表达我们的思想,甚至可以在网站上为自己营造一个家,但是如果你想创造一个更美丽的家,你还需要下一部分的学习.

第二部分 用 CSS 美化你的页面

我们在第一部分已经讲述了 HTML 的基本知识而且能够应用这些知识建立网页,组织我们的内容,接下来的时间里我们的主要任务是让我们的页面打扮的漂亮些.我们将在这一部分学习使用样式表,美化我们的页面.我们会学习样式表的基础知识,处理字体,图文,色彩,背景和定位等,带你进入样式表的世界里漫游.

第十一讲 如何引用样式表

我们在这一讲的主要内容是让大家
了解样式表的优势
认识样式表 学习如何在页面中
引用样式表 如何在样式表中使
用注释

11.1 样式表概述

样式表 (stylesheets) 的技术诞生于 1996 年底,全称是层叠样式表 (Cascading Stylesheets - 简称 CSS),他可以对布局、字体、颜色、背景和其它文图效果实现更加精确的控制。

使用样式表只通过修改一个文件就改变大量的网页的外观和格式,使网页的风格统一使用样式表的具体优势表现在以下几个方面:

格式和结构分离

HTML 没有具体控制网页的格式或外观,浏览器自己决定应该如何显示。层叠样式表通过将定义结构的部分和定义格式的部分分离,使我们能够对页面的布局施加更多的控制。HTML 仍可以保持简单明了的初衷。CSS 代码独立出来从另一角度控制页面外观。

更好的控制页面全局的能力 使用层叠样式表我们可以调整字号,字体,控制边距等等,对页面的全局信息更精确的控制。

可以制作出体积更小下载更快的网页

样式表只是简单的文本,就象 HTML 那样。它不需要图象,不需要执行程序,不需要插件。它就象 HTML 指令那样快。使用串接样式表可以减少表格标签及其它加大 HTML 体积的代码,减少图象用量从而减少文件尺寸。

更快更容易地维护及更新大量的网页 没有样式表时,如果我想更新整个站点中所有主体文本的字体,我必须一页一页地修改

每张网页。样式表的主旨就是将格式和结构分离,利用样式表,我可以将站点上所有的网页都指向单一的一个 CSS 文件,我只要修改 CSS 文件中某一行,那么整个站点都会随之发生变动。

11.2 建立第一张样式表

现在我们就开始制作第一张样式表,我们现在可以写一个基本的 Html 文档.

```
<html>
  <head><title>建立第一张样式表</title></head>
  <body>
    <h4>建立第一张样式表</h4>
    <p>知识创造财富,学习改变人生!</p>
  </body>
```

</html> 我想看一下:[htmlCode建立第一张样式表nocss.HTML](#) 现在,我们看一下如何给它加入样式表。只需在最初的<HTML>和<head>标签之间插入以下 代码:

```
<STYLE TYPE="text/css">
<!--
H1 { color: green; font-size: 37px; font-family: impact }
P { text-indent: 1cm; background: yellow; font-family: courier }
-->
</STYLE>
```

全部代码如下:

```
<html>
  <STYLE TYPE="text/css">
    <!--
      H4 { color: green; font-size: 37px; font-family: impact }
      P { text-indent: 1cm; background: yellow; font-family: courier }
    -->
  </STYLE>
  <head><title>建立第一张样式表</title></head>
  <body>
    <h4>建立第一张样式表</h4>
    <p>知识创造财富,学习改变人生!</p>
  </body>
```

</html> 我想看一下:[htmlCode建立第一张样式表.HTML](#) 打开浏览器我们会发现我们刚刚建立的普通的 Html 文档,在加入样式表后,对字体大小,风格,颜色,背景进行了精确的控制.

我们对上面的样式表代码进行简单的理解:层

叠样式表的核心是规则。最简单的规则就象这样:

```
H4 { color: green }
```

它告诉浏览器将所有<H4></H4>包围的文字以绿色显示,每一条规则包含一个选择及说明。

在上述例子中，H4 就是选择，它是一个附带样式功能的 HTML 标签。说明用于定义实际的样式，包括两部分：属性（本例中即 color）和参数（green）。

任何 HTML 标签都可用作标签。所以你可以将样式表的信息附加到任何要素。从通常的<P>到<CODE>及<TABLE>内容。你甚至可以通过将样式表用于将层叠样式表的属性用于图象。

从我们的第 1 个样式表实例中可以发现，你可以归类样式表的规则。我们将 3 种不同的说明都用于<P>。

与之类似，你也可以归类选择：

H1, P, BLOCKQUOTE { font-family: arial } 这项规则设定所以位于<H1>、<P>和<BLOCKQUOTE>的标签将用 Arial 字体显示。继承性

样式表的规则可从母体延续到子体。下面是一个例子：B { color: blue } 这项规则告诉浏览器将所有之内的文字用蓝色显示。但是在下列情况下，浏览器该如何处理呢？

我的邮箱是:<I>lxz003@126.com</I> 对于<I>标签并没有设定样式，但因为<I>位于之中，所以它将继承母体设定的样式，也以蓝色显示。

11.3 把样式表加入到网页中

我们已经学了将样式表加到网页的一种方法。实际上你可以使用 4 种方法。每种方法都有其不同的优点：

- 将样式表植入 HTML 文件中
- 将样式表链接到 HTML 文件上
- 将样式表输入到 HTML 文件中
- 将样式表加入到 HTML 文件行中

植入样式表：

这就是我们在上面用的方法，所有的样式表信息都列于 HTML 文件的顶部，同<BODY>分列。比如：

```
<html>
  <STYLE TYPE="text/css">
    <!--
      H4 { color: green; font-size: 37px; font-family: impact }
      P { text-indent: 1cm; background: yellow; font-family: courier }
    -->
  </STYLE>
  <head><title>建立第一张样式表</title></head>
  <body>
    <h4>建立第一张样式表</h4>
    <p>知识创造财富,学习改变人生!</p>
  </body>
</html>
```

植入样式表规则后，浏览器在整个 HTML 页面中都执行该规则。如果你想对网页一次

性加入样式表，就可采用该方法。

你可能注意到代码中有两处很奇怪：TYPE="text/css" 属性和注释标签。TYPE="text/css" 设定采用 MIME 类型(MIME, 全称为 "Multipurpose Internet Mail Extensions", 比较确切的中文名称为 "多用途互联网邮件扩展"。它是当前广泛应用的一种电子邮件技术规范), 这样以来, 不支持 CSS 的浏览器可以忽略样式表。注释标签(<!-- and -->)更为重要。有些老的浏览器 (如 MAC 机用的 IE 2.0) 即使在设定了 TYPE="text/css" 属性时也不能忽略样式表继续执行下面的命令, 而且还会显示样式表的代码。而使用注释标签则可以避免发生这种情况。

链接到样式表上

这里是样式表功能发挥得淋漓尽致的地方。你可以将多个 HTML 文件都链接到一个中心样式表文件。这个外部的样式表文件将设定你所有网页的规则。如果你改变样式表文件中的某一细节, 所有页面都会随之改变。如果你维护的站点很大, 则这项功能绝对会有其用武之地。它的使用方法: 产生一个普通的网页, 不是使用<STYLE>规则, 而是在<HEAD>内使用

<LINK>标签:

```
<html>
  <head>
    <title>建立第一张样式表</title>
    <LINK REL=stylesheet HREF="mystyles.css" TYPE="text/css">
  </head>
  <body>
    <h4>建立第一张样式表</h4>
    <p>知识创造财富,学习改变人生!</p>
  </body>
</html>
```

使用链接的样式表时, 你无须使用注释标签, 现在还要生成一个额外的样式表文件, 我们暂时命名为 mystyles.css。文件内容如下:

```
H4 { color: green; font-size: 37px; font-family: impact }
P { text-indent: 1cm; background: yellow; font-family: courier }
```

在 HREF 属性中你可以选择使用绝对或相对 URL。

输入样式表 输入外部样式表的方法同链接的方法类似, 不同之处在于链接法不能同其它方法结合使

用, 但输入法则可以。例如:

```
<HTML>
  <STYLE TYPE="text/css">
  <!--
    @import url(company.css);
    H1 { color: orange; font-family: impact }
  -->
</STYLE>
<HEAD>
  <TITLE>我的样式表</TITLE>
</HEAD>
<BODY>
```

```

    <H1>我的样式表</H1>
    <P>定向培养软件人才,用心打造金领生活!</P>
  </BODY>
</HTML>

```

而其中输入的 company.css 文件内容如下：

```

H1 { color: green; font-family: times }
P { background: yellow; font-family: courier }

```

在本例中，浏览器首先输入 company.css 的规则（@import 必须打头），然后加入移植的规则从而为这个网页产生规则集合。

请注意，对于 H1 在外部样式表文件和植入的样式表中都设定了规则。在两者冲突的情况下，植入的规则此时将占上风。会按照植入的样式表表现网页内容。输入样式表的灵活性可以使你输入无数多个样式表，并可以按照自己的喜好用植入的样式表来覆盖输入的样式表。在行内加入样式

你还可以在 HTML 行中加入样式规则，如下：

```

<HTML>
  <HEAD>
    <TITLE>我的样式表</TITLE>
  </HEAD>
  <BODY>
    <H1 STYLE="color: orange; font-family: impact">定向培养软件人才,用心打造金领生
    活!Go</H1>
    <P STYLE="background: yellow; font-family: courier">北软教育</P>
  </BODY>
</HTML>

```

在这个例子中，你无须在 HTML 顶部加入样表代码。加入行内的样式表属性将使浏览器同样执行样式表规则。该方法不方便之处在于：你必须在每行指中都中加入样式规则，否则下一行时浏览器将转回到文件的缺省设置，加入行内的样式表相比不如植入、链接及输入的样式表那样功能强大，但有时候你会发现它也很有用。

11.4 分类引用

我们先前已经讲过所有的 HTML 标签都可以用作选择，并附加样式表说明。但是还有更复杂的情况，比如，你想让主体文字的 1 段用绿色显示，第 2 段用紫色显示，而第 3 段用灰色显示，这种情况下分类将发挥作用。你可以将段落 P 分成 3 种不同的类别，每一类应用不同的样式表说明。这些规则将以以下方式显示：

```

P.first { color: green }
P.second { color: purple }
P.third { color: gray }

```

你的 HTML 代码可以这样使用样式表：

```

<P CLASS=first>第一段</P>
<P CLASS=second>第二段</P>
<P CLASS=third>第三段</P>

```

你还可以生成不加任何 HTML 标签的分类：

.first { color: green }

这种方式更加灵活，因为现在我们可以将 CLASS=first 用于任何 HTML 标签，而设定的文字将以绿色显示。

11.5 注释

即使是用样式表制作出的非常简练的代码也应该加上注释.例如：

```
P.first { color: green } /* 这里是注释*/  
H1 { text-indent: 10px; font-family: verdana }  
IMG { margin-top: 100px } /* 这里又是一段注释*/
```

第十二讲 字体控制

这一讲我们主要学习字体，即如何控制字号，字模及加入特殊效果。

主要学习内容包括：

- 字体系列
- 字号 字体
- 风格 文字
- 修饰

12.1 按名称调用字体

字体系列就是 CSS 中要使用的一系列字体。其基本语法如下：

```
H2 { font-family: helvetica,impact, sans-serif }
```

这时 Web 浏览器阐释样式表的规则：首先在列表中寻找字体的名称（helvetica），如果在该计算机中安装了这种字体，就使用它。如果没有安装，则移向下一字体（impact），如果这种字体也没有安装，则移向第 3 种字体（sans-serif）。sans-serif 字体参数是浏览器可以依据的最后一种参数，它告诉浏览器使用任何一种缺省 sans-serif 字体（或许就是 Arial）。

将常用字体作为最后的选项列在字体列表中是一种很好的做法。你可以选择以下几种：serif (也可能是 Times)

sans-serif (也可能是 Arial 或者 Helvetica)

cursive (也可能是 Comic Sans)

fantasy (也可能是 Ransom)

monospace (也可能是 Courier)

如果一种字体的名称不只一个单词，如 Gill Sans，在你的 CSS 代码中用引号包含该字体的名称。

```
BODY { font-family: "gill sans", "new baskerville", serif }
```

行内加入的 CSS 规则，使用单引号：

```
<P STYLE="font-family: 'gill sans', 'new baskerville', serif">这里是文本!</P>
```

12.2 字号的控制

在 html 中对字号的控制是很有限的，你只能设定 7 种字号，但是用样式表你可以更自由的设定字号。

使用 font-size 属性，你可以对文字的尺寸进行无限的控制。

points:

```
P { font-size: 16pt }
```

这条代码告诉浏览器以 16 points（点）的尺寸显示<P>。

Points 是确定文字尺寸非常好的单位，因为它在所有的浏览器和操作平台上都适用。

px:

```
P { font-size: 20px }
```

从网页设计的角度来说，pixel (象素) 是一个非常熟悉的单位，它最大的优点就在于所有的操作平台都支持 pixel 单位

12.3 字体风格

利用 CSS 加入斜体字非常简单，而且我们对于加粗字体还有很多选择。

```
H3 { font-style: italic }
```

当浏览器接到应用斜体字的指令后，它将用户的机器中寻找安装的斜体字版本，果不存在斜体字，浏览器将会把正常的字体倾斜显示。

加粗字体

```
P { font-weight: bold }
```

你还可以用数值 100, 200, ...900。正常的非加粗的字体值为 400, 900 是加粗字体的最大值。

12.4 文字修饰 (**text-decoration**)

使用下划线:

```
B { text-decoration: underline }
```

还有其他的修饰内容比如:

overline 给文字上划线

line-through 给文字划出删除线，

none 使得上述效果都不会发生。

none 参数非常有用，你可以使链接的文字不以下划线的形式显示。具体做法：

```
A:link { text-decoration: none }
```

```
A:active { text-decoration: none }
```

```
A:visited { text-decoration: none }
```

12.5 应用举例

代码:

cssStyle.css

```
H3 { font-family: helvetica,impact, sans-serif }
```

```
P.first { font-size: 8pt }
```

```
P.second { font-size: 16pt }
```

```
P.three { font-size: 32pt }
```

```
p.boldWord {font-weight:bold}
```

```
p.boldWord800 {font-weight:100}
```

```
p.italicWord {font-style:italic}
```

```
p.underlineWord {text-decoration: underline }
```

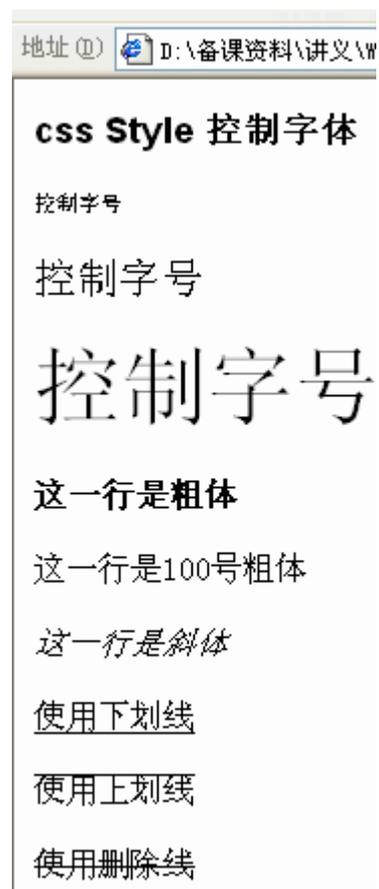
```
p.overlineWord {text-decoration: overline }
```

```
p.delWord {text-decoration: line-through }
```

cssStyle 字体控制.html

```
<Html>
  <head>
    <title>cssStyle 字体控制</title>
    <link rel=stylesheet href="cssStyle.css" type="text/css">
  </head>
  <body>
    <H3>css Style 控制字体</H3>
    <p class="first">控制字号</p>
    <p class="second">控制字号</p>
    <p class="three">控制字号</p>
    <p class="boldWord">这一行是粗体</p>
    <p class="boldWord100">这一行是 100 号粗体</p>
    <p class="italicWord">这一行是斜体</p>
    <p class="underlineWord">使用下划线</p>
    <p class="overlineWord">使用上划线</p>
    <p class="delword">使用删除线</p>
  </body>
</Html>
```

我想看一下:<htmlCode\cssStyle字体控制.HTML>



第十三讲 文字的定位

我们在上面的课程中学习了字体及控制文字的方法.今天我们将学习文字和每行的内容之间是如何定位的。这一讲的学习内容如下：

字间距 行高

文字对齐,缩进

控制边距,边框和空白填充

13.1 字间距

利用字(词)间距属性，你可以在字之间加入更大的距离：

```
H3 { word-spacing: 10px }
```

字母间距:

```
H3 { letter-spacing: 10px }
```

13.2 行高

利用行高我们可以随心所欲地控制行与行之间的垂直距离。

```
B { line-height: 16pt }
```

当你用数字设定行高的时候，浏览器将利用字号来确定行距：它将字号乘以设定的参数值。所以，在下面的例子中，行高将是 24 点。

```
B { font-size: 12pt; line-height: 2 }
```

用比例设定行距

```
B { font-size: 10pt; line-height: 140% }
```

在本例中，行距是长度 10points 的 140%，即 14points。

13.3 文字对齐

利用文字对齐属性，你可以控制段落的水平对齐：

```
H4 { text-align: center }
```

这项属性只用于整块的内容，如<P>、 <H1>-<H6>、<BLOCKQUOTE>和 。

以下为各种选项：

left 指将要素左对齐

right 指将要素右对齐

center 指将要素居中

控制垂直对齐

H4 { vertical-align: top } 垂直对齐可以使你控制文字或替代要素（比如图象）相对于其母体要素的垂直位置。例如，如果你将一个 2x2 象素的 GIF 图象同其母体要素文字的顶部垂直对齐，则该 GIF 图象将在该行文字的顶部显示。

以下为各项参数：

top 将要素顶部同最高的母体要素对齐。

bottom 将要素的底部同最低的母体要素对齐。

text-top 将要素的顶部同母体要素文字的顶部对齐。

text-bottom 将要素的底部同母体要素文字的底部对齐。

baseline 将要素的基准线同母体要素的基准线对齐。

middle 将要素的中点同母体要素的中点对齐。

sub 将要素以下标的形式显示。

super 将要素以上标的形式显示。

13.4 文字缩行

```
P { text-indent: 40% }
```

该属性只应用于整块的内容。你可以使用任何一种熟悉的长度单位设定缩行。你还可以使用比例参数。

例如，本例的缩行为 40% 的比例，即第 1 行的起始位置比其正常位置缩进了 40%。如果你将文字缩行赋予一个负值，则文字第 1 行的起始位置比其正常位置还偏左。

13.5 控制边距,边框和空白填充

我们看一下边距,边框,空白填充的定义:

围绕要素的空白填充 (padding) 围绕空白填充的边框 (border) 围绕边框的边距 (空白 - margin)

图示如下:



控制顶边距、底边距、左边距和右边距

这 4 项属性可以使你控制一个要素的四周的边距。如下：

```
H4 { margin-top: 20px; margin-bottom: 5px; margin-left: 100px; margin-right: 55px }
```

设定边距参数值最明显的方法是利用长度位：px(像素)、pt(点数)等。你也可以用比例值设定边距。控制顶空格填充，底空格填充，左空格填充和右空格填充 空格填充的作用类似于边距控制，你可以设定一个要素的前后左右的空格填充的尺寸。

H4 { padding-top: 20px; padding-bottom: 5px; padding-left: 100px; padding-right: 55px }
 控制位于边距和空格填充之间的边框 控制顶边框宽度，底边框宽度，左边框宽度和
 右边框宽度 你可以控制整个边框的宽度，也可以分别控制每一边的边框宽度：

H4 { border-top-width: 2px; border-bottom-width: 5px; border-left-width: 1px; border-right-width:
 1px }

如果你想使边框四面应用相同的宽度，很简单，使用 border-width 标签就可一次完成，例：

IMG { border-width: 2px }

设置边框颜色

P { border-color: green; border-width: 3px } 如果你想使边框的四边显示不同的颜
 色，你可以在设置中分别列出各种颜色： P { border-color: #666699 #FF0033

#000000 #FFFF99; border-width: 3px }

浏览器将第 1 种颜色理解为顶边框的颜色参数值，第 2 种颜色为右边框，然后是底边框，然
 后是左边框。

边框样式 你还可以设定边框线的样
 式，例如：

P { border-style: double; border-width: 3px }

可以使用的关键字参数值如下：线型和样
 式代码如下：



13.6 综合例子

代码:

cssStyle.css

p.wrodsp10 { word-spacing: 10px }

p.wrodsp20 { word-spacing: 20pt }

p.lettersp10 { letter-spacing: 10pt }

p.lettersp20 { letter-spacing: 20pt }

p.lineH { line-height: 40pt }

h5.left { text-align: left }

h5.right { text-align: right }

h5.center { text-align: center }

```
p.textindent {text-indent:4%}
p.margin { margin-top: 20px; margin-bottom: 5px; margin-left: 100px; margin-right: 55px }
p.padding { padding-top: 20px; padding-bottom: 5px; padding-left: 100px; padding-right: 55px }
p.border { border-color: green; border-style: solid; border-top-width: 2px; border-bottom-width: 5px; border-left-width: 1px; border-right-width: 1px }
p.double { border-color: blue; border-style: double; border-width:2; }
```

cssStyle 字体定位.html

```
<Html>
  <head>
    <title>cssStyle 字体定位</title>
    <link rel=stylesheet href="cssStyle.css" type="text/css">
  </head>
  <body>
    <p class="wrodsp10">word spacing is 10</p>
    <p class="wrodsp20">word spacing is 20</p>
    <p class="lettersp10">letter spacing is 10</p>
    <p class="lettersp20">letter spacing is 20</p>
    <p class="lineH">生活的理想,<br>就是理想的生活!</p>
    <h5 class="left">既然选择了远方,便只顾风雨兼程!</h5>
    <h5 class="center">既然选择了远方,便只顾风雨兼程!</h5>
    <h5 class="right">既然选择了远方,便只顾风雨兼程!</h5>
    <p class="textindent">有一对兄弟,有一天他们出去爬山,然后一起回家。他们的家住在八十层楼,他们一人背着一大包的行李回家,却发现大楼停电了。于是哥哥就说:弟弟,我们一起爬楼梯上去吧。于是他们就一起爬上去。到了二十楼的时候,哥哥又告诉弟弟:包包太重了,我们把它放在二十楼,我们爬上去,明天再下来拿。弟弟说:好。</p>
    <p class="margin">于是他们就把他们的包包放在二十楼,继续往上爬。到了四十楼,弟弟开始抱怨,于是就跟哥哥吵起来了。他们边吵边爬,爬到了六十楼,哥哥就对弟弟说:只剩二十层楼了,我们不要吵了,默默的爬完它吧 !!于是他们就各走各的,终于到了家门口。哥哥就摆出了很帅的姿势:弟弟开门。弟弟就对哥哥说:别闹了,钥匙不是在你那儿。结果,他们把钥匙留在二十楼的包包里了。</p>
    <p class="border">于是他们就把他们的包包放在二十楼,继续往上爬。到了四十楼,弟弟开始抱怨,于是就跟哥哥吵起来了。他们边吵边爬,爬到了六十楼,
```

哥哥就对弟弟说:只剩二十层楼了,我们不要吵了,默默的爬完它吧 !!

</p>

<p class="double">于是他们就把他们的包包放在二十楼,继续往上爬。到了四十楼,弟弟开始抱怨,于是就跟哥哥吵起来了。他们边吵边爬,爬到了六十楼,哥哥就对弟弟说:只剩二十层楼了,我们不要吵了,默默的爬完它吧 !!

</p>

</body>

</html>

我想看一下:[htmlCode\cssStyle字体定位.HTML](#)

```
word spacing is 10
```

```
word spacing is 20
```

```
letter spacing is 10
```

```
letter spacing is 20
```

有一对兄弟,有一天他们出去爬山,然后一起回家。他们的家住在八十层楼,于是哥哥就说:弟弟,我们一起爬楼梯上去吧。于是他们就一起爬上去。到了二十层楼,我们爬上去,明天再下来拿。弟弟说:好。

于是他们就把他们的包包放在二十楼,继续往上爬。到了四十楼,吵边爬,爬到了六十楼,哥哥就对弟弟说:只剩二十层楼了,我们的,终于到了家门口。哥哥就摆出了很帅的姿势:弟弟开门。弟弟果,他们把钥匙留在二十楼的包包里了。

于是他们就把他们的包包放在二十楼,继续往上爬。到了四十楼,弟弟开始抱怨,哥哥就对弟弟说:只剩二十层楼了,我们不要吵了,默默的爬完它吧 !!

于是他们就把他们的包包放在二十楼,继续往上爬。到了四十楼,弟弟开始抱怨,哥哥就对弟弟说:只剩二十层楼了,我们不要吵了,默默的爬完它吧 !!

第十四讲 控制颜色和背景

我们在这一讲的主要焦点将放在颜色和背景的控制上,我们这一讲将学习以下内容:

颜色 背景色 背景图片
的添加和控制

14.1 颜色

颜色属性并不陌生,它的用法很象 HTML 中使用的参数值。B { color: #FF0000 } 这个例子的效果是网页中对加粗字体使用指定的颜色 (#FF0000)显示。

14.2 背景色

利用 CSS 在网页要素后面加入固定的背景色及图象,你甚至可以将要素只加到某一个单词后面。

```
P,yellow { background-color: #FFFF66 }
```

14.3 背景图片

你可以很轻松地将 GIF 或 JPEG 图片设置成背景图片:

```
B { background-image: url(background.gif) }
```

该规则将背景图象加到整个段落之后。你可以将 GIF 图象采用和背景图象通常的设置类似的方法平铺到文字后面,你还可以将背景图象只添加到两个单词后面。

你可以使用 URL 调用一幅图象,你既可以使用相对 URL,例如 images/bg.gif 或完整的 URL,例如

```
http://www.webmonkey.com/images/bg.gif.
```

14.4 控制背景图片

你不仅可以将图象放在要素后面,你还可以精确地控制背景图象的各项设置。你可以决定是否及如何将其平铺,背景图象应该滚动还是保持固定,及将其放在什么位置。背景重复(**background-repeat**)

```
P { background-repeat: no-repeat; background-image: url(
background.gif) }
```

上述规则应用于整个段落。我们以前用过的 GIF 图象出现在文字后面，但由于我们应用了不重复图象 (no-repeat) 规则，所以该图象不会重复平铺在整个段落后面，它只显示一幅单独的图象。

如果你只想让图象垂直或者水平方向平铺，你可以使用 repeat-x 将其水平平铺，用 repeat-y 将其垂直平铺。而 repeat 参数值则将图象从水平和垂直两个方向平铺。固定背景

(background - attachment)

在 HTML 中，背景图象通常会随页面的滚动而一起滚动。但利用 CSS 的固定背景 (background) 属性，你可以建立不滚动的背景图象，页面滚动时，背景图象可以保持固定。BODY { background-attachment: fixed; background-image: url(background.gif) } 这个例子你可以看到页面滚动时，背景仍保持固定。

其参数有两种选择：

scroll 指背景图象随文字内容一起滚动，即通常所见的方式。

fixed 指文字滚动时，背景图象保持固定。背景定位

(background-position) 你可以设定将背景插在什么位置显示。

P { background-position: center bottom; background-image: url (background.gif) }

当上述 CSS 规则应用于本段文字时，背景图象将在本段的中下部开始显示并向右平铺。

设定位置的 3 种方法：

z 关键字参数 (Keyword values)

关键字设定方法简便易用：

top 将背景图象同前景要素的顶部对齐。

bottom 将其同前景要素的底部对齐。

left 将其同左边对齐。

right 将其同右边对齐。

center 将其水平居中 (如果使用在另一关键字前面) 或垂直居中 (如果用在另一关键字后面)。

z 长度参数 长度参数可以使用你对背景图象的位置作出更精确的控制。你可以设定水平和垂直定位 起点，例：

P { background-position: 70px 10px; background-repeat: repeat-y; background-image: url (background.gif) }

将 GIF 图象设置在从左上角起水平 70 象素，垂直 10 象素的位置。由于我设置了 background-repeat: repeat-y，GIF 图象将在文字后垂直平铺。

z 比例值 (Percentage values) 你还可以使用比例

值设定背景图象的位置，例：

P { background-position: 75% 50%; background-image: url (background.gif) }

当这条 CSS 规则应用于本段时，背景图象从水平距离段落右端 70%，垂直距离段落顶部 50% 的位置显示。如果你重新调整浏览器窗口的尺寸，使得段落的显示尺寸变化，背景图象的位置也相应发生变化。

14.5 综合例子

代码:

cssStyle.css

```

/* 颜色和背景*/
b.color{color:blue} p.yellow{background-
color:yellow} p.image{background-
image:url(caomei.jpg)}
p.imagenorep{background-repeat:no-repeat;background-image:url(caomei.jpg)}
body.fiximage{background-attachment: fixed; background-image: url(caomei.jpg)}
body.scrollimage{background-attachment: scroll; background-image: url(caomei.jpg)}

```

cssStyle 颜色和背景.html

```

<Html>
  <head>
    <title>cssStyle 字体定位</title>
    <link rel=stylesheet href="cssStyle.css" type="text/css">
  </head>
  <body>
    <b class="color">既然选择了远方,便只顾风雨兼程!</b>
    <p class="yellow">江雪<br>
    千山鸟飞绝,<br> 万径人踪
    灭.<br> 孤舟蓑笠翁,<br> 独
    钓寒江雪.</p>
    <p class="image">江雪<br>
    千山鸟飞绝,<br> 万径人踪
    灭.<br> 孤舟蓑笠翁,<br> 独
    钓寒江雪.</p>
    <p class="imagenorep">江雪<br>
    千山鸟飞绝,<br> 万径人踪
    灭.<br> 孤舟蓑笠翁,<br> 独钓寒
    江雪.</p>
  </body>
</Html>

```

我想看一下:[htmlCode\cssStyle颜色和背景.HTML](#)

地址 (D) D:\备课资料\讲义\Web 基础_Html_Css_

既然选择了远方,便只愿风雨兼程!



江雪
千山鸟飞绝,
万径人踪灭。
孤舟蓑笠翁,
独钓寒江雪。



江雪
千山鸟飞绝,
万径人踪灭。
孤舟蓑笠翁,
独钓寒江雪。



江雪
千山鸟飞绝,
万径人踪灭。
孤舟蓑笠翁,
独钓寒江雪。

cssStyle 控制图片背景 fix.html

```
<Html>
  <head>
    <title>控制图片固定背景</title>
    <link rel=stylesheet href="cssStyle.css" type="text/css">
  </head>
  <body class="fiximage">
```

<p>有一对兄弟,有一天他们出去爬山,然后一起回家。他们的家住在八十层楼,他们一人背着一大包的行李回家,却发现大楼停电了。于是哥哥就说:弟弟,我们一起爬楼梯上去吧。

于是他们就一起爬上去。

到了二十楼的时候,哥哥

又告诉弟弟:

包包太重了,我们把它放在二十楼,我们爬上去,明天再下来拿。弟弟说:好。于是他们就把他们的包包放在二十楼,继续往上爬。

到了四十楼,弟弟开始抱怨,于是就跟哥哥吵起来了。他们边吵边爬,爬到了六十楼,哥哥就对弟弟说:只剩二十层楼了,我们不要吵了,默默的爬完它吧!!于是他们就各走各的,终于到了家门口。哥哥就摆出了很帅的姿势:弟弟开门。

弟弟就对哥哥说:别闹了,钥匙不是在你那儿。

结果,他们把钥匙留在二十楼的包包里了。

这个故事其实在反映我们的人生。

有很多人在二十岁以前是活在家人的期望和老师的期许之下,背负着很多的压力;在二十岁之后离开了众人的压力,开始满腔的热血,开始有很多的梦想去完成;可是工作了二十年后,开始发觉工作不如意...于是就开始抱怨老板、抱怨公司、抱怨社会、抱怨政府。就在这抱怨中又渡过了二十年。于是告诉自己,六十岁了没什么好抱怨的了,就默默的走完自己的这些年吧。到了八十岁快要死掉的前夕,才想起自己好像有什么事还没完成 原来,他二十岁的梦想还没有完成。
所以,希望现在的你充满着理想...充满着热情去做自己想做的事。人生短短数十载,也没有什么好担心的不是吗?尤其是在经历那么多的遭遇,更应该能体会,把握现在。
记得吗?今天就是礼物!</p>

</body>

我想看一下:[htmlCode\cssStyle控制背景图片fix.HTML](#) 我们在上面的代码中修改以下可以实现背景的滚动body class="scrollimage">我想看一下:[htmlCode\cssStyle控制背景图片scroll.HTML](#)

第十五讲 要素的定位

在 html 中定位文本和图片是很困难的,但在 css 样式表中进行要素的定位是很方便的.所有我们这一讲主要讲述以下内容:

绝对定位和相对定位 宽度,高度,可视性控制

15.1 绝对定位

定位 (左边和顶部)

```
H4 { position: absolute; left: 100px; top: 43px }
```

这项 CSS 规则让浏览器将<H4>的起始位置精确地定在距离浏览器左边 100 像素,距离其顶部 43 像素的位置。注意这里唯一设置了的是左边和顶部,也就是说,文字将从左到右,从上到下载入浏览窗口。

15.2 相对定位

绝对定位使你能精确地定位要素在页面的独立位置,而不考虑页面其它要素的定位设置。相对定位指你所定位的要素的位置相对于在文件中所分配的位置。例:

```
I { position: relative; left: 40px; top: 10px }
```

相对定位的关键在于定位了的要素的位置是相对于它通常应在的位置进行定位.如果你停止使用相对定位,则文字的显示位置将恢复正常。

15.3 控制要素的宽度

```
DIV { position: absolute; left:200px; top: 40px; width: 150px }
```

浏览器接到这项规则时,它将文字按照规则规定的效果显示,还将段落的最大水平尺寸限制在 150 像素。

IE4 中,这项属性还可用于图象.你可以通过宽度设置人为地拉宽或压缩图象。

15.4 控制高度

高度应该和宽度的设置类似,只不过是在垂直方向上:

```
DIV {position:absolute;left:200px;top:40px;height:150px}
```

15.5 可视性

利用 CSS,你可以隐藏要素,使其在页面上看不见.

```
H4 { visibility: hidden }
```

选项:

visible 使要素可以被看见

hidden 使要素被隐藏

inherit 指它将继承母体要素的可视性设置.

当一个要素被隐藏后,它仍然要占据浏览器窗口中的原有空间。所以,如果你将文字包围在一幅被隐藏的图象周围,那么,其显示效果将是文字包围着一块空白区域.

第三部分 使用 JavaScript 和用户交互

通过超文本(Hyper Text)和超媒体(Hyper Media)技术结合超链接(Hyper link)的连接功能将各种信息组织成网络结构(web), 构成网络文档(Document), 实现 Internet 上的“漫游”。

通过 HTML 符号的描述就可以实现文字、表格、声音、图像、动画等多媒体信息的检索。然而采用这种超链技术存在有一定的缺陷, 那就是它只能提供一种静态的信息资源, 缺少动态的客户端与服务器端的交互。虽然可通过 CGI (Common Gateway Interface)通用网关 接口

实现一定的交互, 但由于该方法编程较为复杂, 因而在一段时间防碍了 Internet 技术的发展。而 JavaScript 的出现, 无疑为 Internet 网上用户带来了一线生机。可以这样说, JavaScript 的出现是时代的需求, 是当今的信息时代造就了 JavaScript。

JavaScript 的出现, 它可以使得信息和用户之间不仅只是一种显示和浏览的关系, 而是实现了一种实时的、动态的、可交式的表达能力。从而基于 CGI 静态的 HTML 页面将被可提供动态实时信息, 并对客户操作进行反应的 Web 页面的取代。JavaScript 脚本正是满足这种需求而产生的语言。它深受广大用户的喜爱的欢迎。它是众多脚本语言中较为优秀的一种, 它与 WWW 的结合有效地实现了网络计算和网络计算机的蓝图。

第十六讲 JavaScript 语言概况

我们在这一讲会向你介绍:

什么是 **JavaScript**

JavaScript 的特点以及与其他语言的区别

JavaScript 的运行环境 建立

一个简单的 **JavaScript**

16.1 什么是 JavaScript

JavaScript 是一种基于对象(Object)和事件驱动(Event Driven)并具有安全性能的脚本语言。使用它的目的是与 HTML 超文本标记语言、Java 脚本语言 (Java 小程序) 一起实现在一个 Web 页面中链接多个对象, 与 Web 客户交互作用。从而可以开发客户端的应用程序等。它是通过嵌入或调入在标准的 HTML 语言中实现的。它的出现弥补了 HTML 语言的缺陷, 它是 Java 与 HTML 折衷的选择, 具有以下几个基本特点:

z 它是一种脚本编写语言

JavaScript 是一种脚本语言, 它采用小程序段的方式实现编程。像其它脚本语言一样, JavaScript 同样已是一种解释性语言, 它提供了一个易的开发过程。它的基本结构形式与 C、C++、VB、Delphi 十分类似。但它不像这些语言一样, 需要先编译, 而是在程序运行过程中被逐行地解释。它与 HTML 标识结合在一起, 从而方便用户的使用操作。

z 基于对象的语言

JavaScript 是一种基于对象的语言, 同时可以看作一种面向对象的。这意味着它能运用自己已经创建的对象。因此, 许多功能可以来自于脚本环境中对象的方法与脚本的相互作用。

z 简单性

JavaScript 的简单性主要体现在: 首先它是一种基于 Java 基本语句和控制流之上的简单而紧凑的设计, 从而对于学习 Java 是一种非常好的过渡。其次它的变量类型是采用弱类型, 并未使用严格的数据类型。

z 安全性

JavaScript 是一种安全性语言, 它不允许访问本地的硬盘, 并不能将数据存入到服务器上, 不允许对网络文档进行修改和删除, 只能通过浏览器实现信息浏览或动态交互。从而有效地防止数据的丢失。

z 动态性

JavaScript 是动态的, 它可以直接对用户或客户输入做出响应, 无须经过 Web 服务程序。它对用户的反映响应, 是采用以事件驱动的方式进行的。所谓事件驱动, 就是指在主页 (Home Page) 中执行了某种操作所产生的动作, 就称为“事件” (Event)。比如按下鼠标、移动窗口、选择菜单等都可以视为事件。当事件发生后, 可能会引起相应的事件响应。

z 跨平台性

JavaScript 是依赖于浏览器本身, 与操作环境无关, 只要能运行浏览器的计算机, 并支持 JavaScript 的浏览器就可正确执行。从而实现了“编写一次, 走遍天下”的梦想。

综上所述 JavaScript 是一种新的描述语言, 它可以被嵌入到 HTML 的文件之中。

JavaScript 语言可以做到回应使用者的需求事件 (如: form 的输入), 而不用任何的网路来回传输资料, 所以当一位使用者输入一项资料时, 它不用经过传给服务端 (server) 处理, 再传回来的过程, 而直接可以被客户端 (client) 的应用程序所处理。

JavaScript 和 Java 很类似, 但到底并不一样! Java 是一种比 JavaScript 更复杂许多的程式语言, 而 JavaScript 则是相当容易了解的语言。JavaScript 创作者可以不那么注重程序技巧, 所以许多 Java 的特性在 Java Script 中并不支持。

16.2 JavaScript 和 Java 的区别

虽然 JavaScript 与 Java 有紧密的联系, 但却是两个公司开发的不同的两个产品。Java 是 SUN 公司推出的新一代面向对象的程序设计语言, 特别适合于 Internet 应用程序开发; 而 JavaScript 是 Netscape 公司的产品, 其目的是为了扩展 Netscape Navigator 功能, 而开发的一种可以嵌入 Web 页面中的基于对象和事件驱动的解释性语言。它的前身是 Live Script; 而 Java 的前身是 Oak 语言。

下面对两种语言间的异同作如下比较:

z 基于对象和面向对象

Java 是一种真正的面向对象的语言, 即使是开发简单的程序, 必须设计对象。

JavaScript 是种脚本语言, 它可以用来制作与网路无关的, 与用户交互作用的复杂软件。它是一种基于对象 (Object Based) 和事件驱动 (Event Driver) 的编程语言。因而它本身提供了非常丰富的内部对象供设计人员使用。

z 解释和编译

两种语言在其浏览器中所执行的方式不一样。Java 的源代码在传递到客户端执行之前, 必须经过编译, 因而客户端上必须具有相应平台上的仿真器或解释器, 它可以通过编译器或解释器实现独立于某个特定的平台编译代码的束缚。

JavaScript 是一种解释性编程语言, 其源代码在发往客户端执行之前不需经过编译, 而是将文本格式的字符代码发送给客户端由浏览器解释执行。

z 强变量和弱变量 两种语言所采取的变

量是不一样的。

Java 采用强类型变量检查, 即所有变量在编译之前必须作声明。如:

```
int x;
String y;
x=1234;
x=4321;
```

其中 X=1234 说明是一个整数, Y=4321 说明是一个字符串。

JavaScript 中变量声明, 采用其弱类型。即变量在使用前不需作声明, 而是解释器在运行时检查其数据类型, 如:

```
x=1234;
y = "4321";
```

前者说明 x 为其数值型变量, 而后者说明 y 为字符型变量。

z 代码格式不一样

Java 是一种与 HTML 无关的格式, 必须通过像 HTML 中引用外媒体那么进行装载, 其代码以字节代码的形式保存在独立的文档中。

JavaScript 的代码是一种文本字符格式, 可以直接嵌入 HTML 文档中, 并且可动态装载。编写 HTML 文档就像编辑文本文件一样方便。

- z 嵌入方式不一样
在 HTML 文档中,两种编程语言的标识不同,JavaScript 使用<Script>...</Script>来标识,而 Java 使用<applet>...</applet>来标识。
- z 静态绑定和动态绑定
Java 采用静态绑定,即 Java 的对象引用必须在编译时的进行,以使编译器能够实现强类型检查。
JavaScript 采用动态绑定,即 JavaScript 的对象引用在运行时进行检查。

16.3 JavaScript 程序运行环境

软件环境:

操作系统:Windows 95/98 或 Windows NT.

浏览器:Netscape Navigator x.0 或 Internet Explorer x.0.

编辑工具:用于编辑 HTML 文档的字符编辑器(WS、WPS、Notepad、WordPad 等)或 HTML 文档编辑器。硬件配置:(硬件配置要求很低,这里不再描述)

16.4 编写第一个 JavaScript 程序

下面我们通过一个例子,编写第一个 JavaScript 程序。通过它可说明 JavaScript 的脚本是怎样被嵌入到 HTML 文档中的。

代码:

myFirstJs.HTML

```
<html>
  <head>
    <title>我的第一个 Javascript</title>
    <Script Language ="JavaScript">
      // 下面是 JavaScript 代码
      alert("我的第一 JavaScript!麻烦你点一下确定按钮!");
      alert("叫你按你就按呀!真听话!");
      alert("还在按呀!"); alert("不和你玩了!886");
    </Script>
  </head>
  <body>
    我的第一个 Javascript!
  </body>
</html>
```

我想看一下:[jsCode\myFirstJs.HTML](#)



说明：

myFirstJs.html 是 HTML 文档,其标识格式为标准的 HTML 格式,如同 HTML 标识语言一样,JavaScript 程序代码是一些可用字处理软件浏览的文本,它在描述页面的 HTML 相关区域出现。

JavaScript 代码由 `<Script Language = "JavaScript">...</Script>` 说明。在标识 `<Script Language = "JavaScript">...</Script>` 之间就可加入 JavaScript 脚本。

`alert()` 是 JavaScript 的窗口对象方法,其功能是弹出一个具有 OK 对话框并显示 () 中的字符串。

通过 `<!-- .../-->` 标识说明:若不认识 JavaScript 代码的浏览器,则所有在其中的标识均被忽略;若认识,则执行其结果。使用注释这是一个好的编程习惯,它使其他人可以读懂你的语言。

JavaScript 以 `</Script>` 标签结束。

第十七讲 JavaScript 基本数据结构

JavaScript 提供脚本语言的编程与 Java 非常相似,也去掉了指针的概念,并提供了功能强大的类库.对于已经具备 Java 或 C 语言的人来说,学习 JavaScript 脚本语言是一件非常轻松愉快的事。

我们在这一讲主要讲述以下内容:

如何嵌入 **JavaScript** 代码 基本的
数据类型
变量和表达式

17.1 JavaScript 代码的加入

JavaScript 的脚本包括在 HTML 中,它成为 HTML 文档的一部分。与 HTML 标识相结合,构成了一个功能强大的 Internet 网上编程语言。可以直接将 JavaScript 脚本加入文档:

```
<Script Language ="JavaScript">
```

```
JavaScript 语言代码;
```

```
JavaScript 语言代码;
```

```
.....
```

```
</Script>
```

说明:

通过标识<Script>...</Script>指明 JavaScript 脚本源代码将放入其间。

通过属性 Language ="JavaScript"说明标识中是使用的何种语言,这里是 JavaScript 语言,表示在 JavaScript 中使用的语言。

下面是将 JavaScript 脚本嵌入到 Web 文档中的例子:

```
<HTML>
```

```
<Head>
```

```
<Script Language ="JavaScript">
```

```
document.write("document.write()的说明:<br>document 是指当前的 html 文档,使用  
JavaScript 脚本中的 document 对象的 write 方法,<br>可以向 html 文档中添加信息!");
```

```
document.close();
```

```
</Script>
```

```
</Head>
```

```
<body>
```

```
<p>上面的文字是用 JavaScript 加入的.</p>
```

```
</body>
```

```
</HTML>
```

我想看一下:<jsCode\js172.HTML>

地址 (D)  D:\备课资料\讲义\Web_基础_Html_Css_JS\jsCode\js172.HTML

document.write()的说明:
document是指当前的html文档,使用JavaScript脚本中的
可以向html文档中添加信息!

上面的文字是用JavaScript加入的.

说明:

Document.write()是文档对象的输出函数,其功能是将括号中的字符或变量值输出到窗口;document.close()是将输出关闭。

可将<Script>...</Script>标识放入 head>.. </Head> 或<Body> ...</Body>之间。将JavaScript标识放置<Head>... </Head>在头部之间,使之在主页和其余部分代码之前装载,从而使代码的功能更强大;可以将JavaScript标识放置在<Body>... </Body>主体之间以实现某些部分动态地创建文档。

我们也可以把JavaScript代码保存在独立的文件中(扩展名为.js),在Html中引入这个文件:

代码:

importJs.js

//嵌入的Js代码

```
document.write("<h4>");
document.write("人活的是一种心态!");
document.write("</h4>");
```

importJs.html

```
<html>
<head>
</head>
<body>
  <script src="importJs.js">
  </script>
  <p>
    上面的一段话是 importJs.js 文件写入的信息!
  </p>
</body>
</html>
```

注意:

在js代码中就不要使用<script></script>去包含Js代码,因为在引用的Html中已经包含了插入Js的标记。

我想看一下:[jsCode\importJs.HTML](#)

17.2 注释

像其他所有语言一样,JavaScript的注释在运行时也是被忽略的。注释只给程序员提供消息。

JavaScript注释有两种:单行注释和多行注释。单行注释用双反斜杠“//”表示。当一行代码有“//”,那么,“//”后面的部分将被忽略。而多行注释是用“/*”和“*/”括起来的

一行到多行文字。程序执行到 “/*” 处，将忽略以后的所有文字，直到出现 “*/” 为止。

例如：

```
//这里是单行注释
/* 下面这些内容,是多
   行注释的情况,和
   Java 语言类似.
*/
```

17.3 基本数据类型

JavaScript 脚本语言同其它语言一样，有它自身的基本数据类型、表达式和算术运算符以及程序的基本框架结构。JavaScript 提供了四种基本的数据类型用来处理数字和文字。基本数据类型

在 JavaScript 中四种基本的数据类型：数值（整数和实数）、字符串型（用 “” 号或 “” 括起来的字符或数值）、布尔型（使 True 或 False 表示）空值。

在 JavaScript 的基本类型中的数据可以是常量，也可以变量。由于 JavaScript 采用弱类型的形式，因而一个数据的变量或常量不必首先作声明，而是在使用或赋值时确定其数据的类型的。当然也可以先声明该数据的类型，它是通过在赋值时自动说明其数据类型的。常量

整型常量

JavaScript 的常量通常又称字面常量，它是不能改变的数据。其整型常量可以使用十六进制、八进制和十进制表示其值。

实型常量

实型常量是由整数部分加小数部分表示，如 12.32、193.98。可以使用科学或标准方法表示：5E7、4e5 等。

布尔值

布尔常量只有两种状态：True 或 False。它主要用来说明或代表一种状态或标志，以说明操作流程。它与 C++ 是不一样的，C++ 可以用 1 或 0 表示其状态，而 JavaScript 只能用 True 或 False 表示其状态。

字符型常量

使用单引号（'）或双引号（"）括起来的一个或几个字符。如 "This is a book of JavaScript"、"3245"、"ewrt234234" 等。

空值

JavaScript 中有一个空值 null，表示什么也没有。如试图引用没有定义的变量，则返回一个 Null 值。

转义字符

由于一些字符在屏幕上不能显示，或者 JavaScript 语法上已经有了特殊用途，在要用这些字符时，就要使用“转义字符”。转义字符用斜杠 “\” 开头：\ 单引号、\ 双引号、\n 换行符、\r 回车（以上只列出常用的转义字符）。

于是，使用转义字符，就可以做到引号多重嵌套：Micro 说：“这里是”JavaScript 教程“。”

NaN

“Not a Number”。出现这个数值比较少见，以至于我们可以不理它。当运算无法返回正确的数值时，就会返回“NaN”值。NaN 值非常特殊，因为它“不是数字”，所以任何数跟它都不相等，甚至 NaN 本身也不等于 NaN。

17.4 变量

变量的主要作用是存取数据、提供存放信息的容器。对于变量必须明确变量的命名、变量的类型、变量的声明及其变量的作用域。

变量的命名

JavaScript 中的变量命名同其计算机语言非常相似，这里要注意以下两点：

- z 必须是一个有效的变量，即变量以字母开头，中间可以出现数字如 test1、text2 等。除下划线 (-) 作为连字符外，变量名称不能有空格、(+)、(-)、(,) 或其它符号。
- z 不能使用 JavaScript 中的关键字作为变量。

在 JavaScript 中定义了 40 多个类键字，这些键是 JavaScript 内部使用的，不能作为变量的名称。如 Var、int、double、true 不能作为变量的名称。

变量的类型

在 JavaScript 中，变量可以用命令 Var 作声明：

```
var mytest;
```

该例子定义了一个 mytest 变量。但没有赋予它的值。

```
Var mytest="This is a book"
```

该例子定义了一个 mytest 变量，同时赋予了它的值。

在 JavaScript 中，变量以可以不作声明，而在使用时再根据数据的类型来确其变量的类型。

如：

```
x=100
```

```
y="125"
```

```
xy= True
```

```
cost=19.5 等。
```

其中 x 整数，y 为字符串，xy 为布尔型，cost 为实型。

变量的声明及其作用域

JavaScript 变量可以在使用前先作声明，并可赋值。通过使用 var 关键字对变量作声明。对变量作声明的最大好处就是能及时发现代码中的错误；因为 JavaScript 是采用动态编译的，而动态编译是不易发现代码中的错误，特别是变量命名的方面。

对于变量还有一个重要性——那就是变量的作用域。在 JavaScript 中同样有全局变量和局部变量。全局变量是定义在所有函数体之外，其作用范围是整个函数；而局部变量是定义在函数体之内，只对其该函数是可见的，而对其它函数则是不可见的。

17.5 表达式和运算符

表达式

在定义完变量后，就可以对它们进行赋值、改变、计算等一系列操作，这就要通过表达式来完成，可以说表达式是变量、常量、布尔及运算符的集合，因此表达式可以分为算术表述式、字串表达式、赋值表达式以及布尔表达式等。

运算符

运算符:完成操作的一系列符号,在 JavaScript 中有算术运算符,如+、-、*、/等;有比较运算符如!=、==等;有逻辑布尔运算符如!(取反)、|(或)、||(或);有字符串运算符如+、+=等。

(1)算术运算符

+ (加)、- (减)、* (乘)、/ (除)、% (取模)、|(按位或)、&(按位与)、<< (左移)、>> (右移)、>>> (右移,零填充)。

-(取反)、~(取补)、++(递加1)、--(递减1)。

(2)比较运算符

比较运算符它的基本操作过程是,首先对它的操作数进行比较,尔后再返回一个 true 或 false 值,有 8 个比较运算符:

<(小于)、>(大于)、<=(小于等于)、>=(大于等于)、==(等于)、!=(不等于)。

(3)布尔逻辑运算符

&&(与),|(或),!(非)

(4)位运算符

~(取补),&(与),|(或),^(异或)

(5)三元操作符 主要格式

如下:操作数?结果1:

结果2

若操作数的结果为真,则表述式的结果为结果1,否则为结果2。

17.6 综合例子

使用变量输出文本的例子

代码:

```
<html>
  <body>
    <script type="text/javascript">
      var myword = "年年岁岁花相似,岁岁年年人不同!"
      document.write("<h4>"+myword+"</h4>")
    </script>
    <p>使用变量在 Html 文档中输出信息的例子</p>
  </body>
</html>
```

我想看一下:[jsCode\useVar.HTML](#)

下面是计算表达式运算的综合例子:

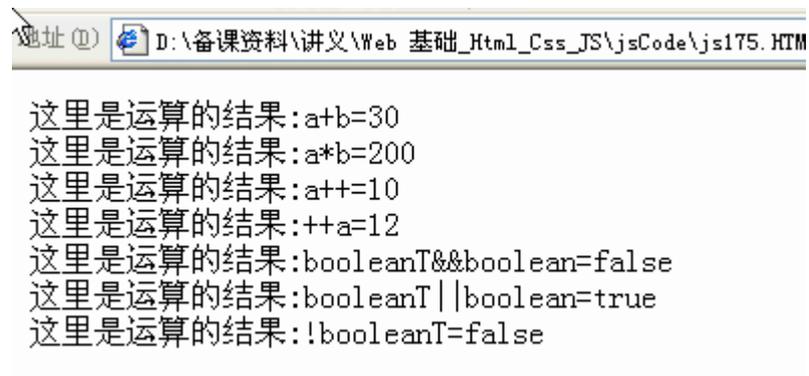
代码:

```
<html>
  <head>
    <title>使用运算符的例子</title>
    <script Language="JavaScript">
      //定义全局整型变量
      var a=10;
      var b=20;
```

```
//定义全局字符串
var c="这里是运算的结果:";
//定义全局布尔变量
var booleanT=true;
var booleanF=false;
//定义一个计算函数
function calculate(){
    //定义局部变量
    var temp;
    var d=a+b;
    e=a*b;
    f="a+b=";
    g="a*b=";
    temp=c+f;
    document.write(temp);
    document.write(d+"<br>");
    temp=c+g;
    document.write(temp);
    document.write(e);
    document.write("<br>");
    document.write(c);
    document.write("a++=");
    document.write(a++);
    document.write("<br>");
    document.write(c);
    document.write("++a=");
    document.write(++a);
    document.write("<br>");
    document.write(c);
    document.write("booleanT&&boolean=");
    document.write(+booleanT&&booleanF);
    document.write("<br>");
    document.write(c);
    document.write("booleanT||boolean=");
    document.write(booleanT||booleanF);
    document.write("<br>");
    document.write(c);
    document.write("!booleanT=");
    document.write(!booleanT);
    //关闭文档
    document.close();
}
//调用计算函数
calculate();
```

```
</script>
</head>
<body>
</body>
</html>
```

我想看一下:<jsCode\js175.HTML>



第十八讲 JavaScript 程序构成

JavaScript 脚本语言的基本构成是由控制语句、函数、对象、方法、属性等,来实现编程的。我们在这一讲主要讲述以下内容:

流程控制语句
函数 事件驱动和事件
处理

18.1 流程控制语句

z if 条件语句

基本格式 if (表
述式) 语句段

1 ;

..... else 语

句段 2 ;

.....

功能：若表达式为 true，则执行语句段 1；否则执行语句段 2。

说明：

if -else 语句是 **JavaScript** 中最基本的控制语句，通过它可以改变语句的执行顺序。表达式中必须使用关系语句，来实现判断，它是作为一个布尔值来估算的。

它将零和非零的数分别转化成 **false** 和 **true**..

若 **if** 后的语句有多行，则必须使用花括号将其括起来

最简单的 *if* 语句的使用

代码:

```
<html>
```

```
  <body>
```

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

```
      var d = new Date()//获得日期
```

```
      var time = d.getHours()//获得小时
```

```
      if (time < 10)
```

```
      {
```

```
        document.write("<b>早上好</b>")
```

```
      }
```

```
      else
```

```
        document.write("<b>希望你一天快乐!</b>");
```

```
    </script>
```

```
  <p>
```

```
    这是一个最简单的 if 语句!
```

```
  </p>
```

```
  <p>
```

```
    如时间在 10 点以前会显示"早上好"!
```

```
  </p>
```

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

我想看一下:[jsCode\ifStatement.HTML](#)

希望你一天快乐!

这是一个最简单的if语句!

如时间在10点以前会显示“早上好”!

z For 循环语句 基本格式 for (初始化,条件,增量)

```
{
    语句集 ;
}
```

功能：实现条件循环，当条件成立时，执行语句集，否则跳出循环体。

说明：

初始化参数告诉循环的开始位置，赋予变量的初值。条件：是用于判别循环停止时的条件。若条件满足，则执行循环体，否则跳出。增量：主要定义循环控制变量在每次循环时按什么方式变化。三个主要语句之间，必须使用逗号分隔。

下面是一个 *for* 循环的例子：

```
<html>
  <head>
    <title>for 循环的例子</title>
    <script type="text/javascript">
      for (i = 1; i <= 6; i++)
      {
        document.write("<h" + i + ">This is header " + i)
        document.write("</h" + i + ">")
      }
    </script>
  </head>
  <body>
  </body>
```

</html> 我想看一

下:[jsCode\js184.HTML](#) z

while 循环

基本格式

while (条件)

```
{
    语句集 ;
```

```
}

```

该语句与 for 语句一样，当条件为真时，重复循环，否则退出循环。

for 与 while 语句

两种语句都是循环语句，使用 for 语句在处理有关数字时更易看懂，也较紧凑；而 while 循环对复杂的语句效果更特别。

while 循环的例子

代码:

```
<html>
  <head>
    <title>while 循环的例子</title>
    <script type="text/javascript">
      i = 0;
      while (i <= 5)
      {
        document.write("The number is " + i);
        document.write("<br>");
        i++;
      }
    </script>
  </head>
  <body>
  </body>
</html>
```

我想看一下:[jsCode\js185.HTML](#)

z do... while 循环

先执行循环后判断条件,循环体至少会执行一次,和 Java 中类似不在讲述.

使用 *do...while* 循环的例子

代码:

```
<html>
  <body>
    <script type="text/javascript">
      i = 0
      do
      {
        document.write("The number is " + i)
        document.write("<br>")
        i++
      }
      while (i <= 5)
    </script>

    <p>do...while 循环的例子,至少要执行一次</p>
  </body>
</html>
```

我想看一下:[jsCode\dowhile.HTML](#)

z break 和 continue 语句

与 Java 语言相同,使用 break 语句使得循环从 For 或 while 中跳出,continue 使得跳过循环内剩余的语句而进入下一次循环。

z switch 语句 如果要把某些数据分类,例如,要把学生的成绩按优、良、中、差分类,我们可能会用 if 语句:

```
if (score >= 0 && score < 60) {
    result = 'fail';
} else if (score < 80) {
    result = 'pass';
} else if (score < 90) {
    result = 'good';
} else if (score <= 100) {
    result = 'excellent';
} else {
    result = 'error';
}
```

看起来没有问题,但使用太多的 if 语句的话,程序看起来有点乱。switch 语句就是解决这种问题的最好方法。

```
switch (e) {
    case r1: (注意:冒号)
        ...
        [break;]
    case r2:
        ...
        [break;]
    ...
    [default:
        ...]
}
```

这一大段的作用是:计算 e 的值(e 为表达式),然后跟下边“case”后的 r1、r2.....比较,当找到一个相等于 e 的值时,就执行该“case”后的语句,直到遇到 break 语句或 switch 段落结束(“}”)。如果没有一个值与 e 匹配,那么就执行“default:”后边的语句,如果没有 default 块,switch 语句结束。

上边的 if 段用 switch 改写就是:

```
switch (parseInt(score / 10)) {
    case 0:
    case 1:
    case 2:
    case 3:
    case 4:
    case 5:
        result = 'fail';
}
```

```

    break;
case 6:
case 7:
    result = 'pass';
    break;
case 8:
    result = 'good';
    break;
case 9:
    result = 'excellent';
    break;
default:
    if (score == 100)
        result = 'excellent';
    else
        result = 'error';
}

```

其中 `parseInt()`方法的作用是取整。最后 `default` 段用的 `if` 语句，是为了不把 100 分当错误论 (`parseInt(100 / 10) == 10`)。

18.2 函数

z JavaScript 函数的定义

```

function 函数名 ( 参数 ) {
    函数体;
    return 表达式;
}

```

说明：

函数由关键字 **function** 定义(关键字为小写) 参数列表，是传递给函数使用或操作的值，其值可以是常量，变量或其它表达式 通过指定函数名（实参）来调用一个函数

使用 **Return** 将值返回 函

数名对大小写是敏感的 z

函数中的形式参数

在函数的定义中，我们看到函数名后有参数列表，这些参数变量可能是一个或几个。那麽怎样才能确定参数变量的个数呢？在 JavaScript 中可通过 `arguments.Length` 来检查参数的个数。例：

```

<html>
  <head>
    <title>确定参数个数</title>
    <Script Language="JavaScript">
      function TestArgLen(a,b,c,d)
      {
        len= TestArgLen.arguments.length;

```

```

        if(len>0)
        {
            alert("a="+a);
        }
        if(len>1)
        {
            alert("b="+b);
        }
        if(len>2)
        {
            alert("c="+c);
        }
        if(len>3)
        {
            alert("d="+d);
        }
        if(len>4)
        {
            alert("e="+e);
        }
    }
TestArgLen(5,7,9);
    document.write("运行结束!");
    document.close();

```

```
</Script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html> 我想看一
```

下:[jsCode\js182.HTML](#) z

函数应用举例 最简单的函数例

子

代码:

```
<html>
```

```
<head>
```

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

```
function myfunction()
```

```
{
```

```
    alert("你好!这里函数调用的运行结果!")
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<form>
```

```

onclick="myfunction()"
value="调用函数">
</form>

```

</body>

我想看一下:[jsCode\sampleFunction.HTML](#)

带参数的函数调用

代码:

```

<html>
  <head>
    <title>使用函数的例子</title>
    <script type="text/javascript">
      function welcome(txt)
      {
        alert(txt)
      }
    </script>
  </head>
  <body>
    <p>
      欢迎光临
    </p>
    <form>
      <input type="button" onclick="welcome('早上好!')" value="现在是早上">
      <input type="button" onclick="welcome('晚上好!')" value="现在是晚上">
    </form>
  </body>

```

</html> 我想看一

下:[jsCode\js186.HTML](#) 带返回

值的函数

代码:

```

<html>
  <head>
    <script type="text/javascript">
      function total(numberA,numberB)
      {
        return numberA + numberB
      }
    </script>
  </head>
  <body>
    <script type="text/javascript">
      document.write("2+3=");
      document.write(total(2,3))
    </script>

```

```

</script>
<p>带返回值的函数调用</p>
</body>
</html>
我想看一下:jsCode\functionReturn.HTML

```

18.3 事件驱动及事件处理

z 基本概念

JavaScript 是基于对象(object-based)的语言。这与 Java 不同,Java 是面向对象的语言。而基于对象的基本特征,就是采用事件驱动(event-driven)。它是在用形界面的环境下,使得一切输入变化简单化。通常鼠标或热键的动作我们称之为事件(Event),而由鼠标或热键引发的一连串程序的动作,称之为事件驱动(Event Driver)。而对事件进行处理程序或函数,我们称之为事件处理程序(Event Handler)。

z 事件处理程序

在 JavaScript 中对对象事件的处理通常由函数(Function)担任。其基本格式与函数全部一样,可以将前面所介绍的所有函数作为事件处理程序。格式如下:

```

function 事件处理名(参数列表)
{
    事件处理语句集;
    .....
}

```

z 事件驱动

JavaScript 事件驱动中的事件是通过鼠标或热键的动作引发的。它主要有以下几个事件:

onblur 事件 发生在窗口失去焦点的时候。

应用于: window 对象

onchange 事件 发生在文本输入区的内容被更改,然后焦点从文本输入区移走之后。捕捉此事件主要用

于实时检测输入的有效性,或者立刻改变文档内容。应用于:

Password 对象; Select 对象; Text 对象; Textarea 对象

onclick 事件 发生在对象被单击的时候。单击是指鼠标停留在对象上,按下鼠标键,没有移动鼠标而

放开鼠标键这一个完整的过程。

一个普通按钮对象(Button)通常会有 onclick 事件处理程序,因为这种对象根本不能从用户那里得到任何信息,没有 onclick 事件处理程序就等于摆设。

按钮上添加 onclick 事件处理程序,可以模拟“提交按钮”,方法是:在事件处理程序中更改表单的 action, target, encoding, method 等一个或几个属性,然后调用表单的 submit()方法。

在 Link 对象的 onclick 事件处理程序中返回 false 值(return false),能阻止浏览器打开此连接。即,如果有一个这样的连接:Go!,那么无论用户怎样点击,都不会去到 www.abc.com 网站,除非用户禁止浏览器运行 JavaScript。

应用于: Button 对象; Checkbox 对象; Image 对象; Link 对象; Radio 对象; Reset 对

象；Submit 对象

onerror 事件

发生在错误发生的时候。它的事件处理程序通常就叫做“错误处理程序”(Error Handler)，用来处理错误。要忽略一切错误，就使用：

```
function ignoreError() {
    return true;
}
```

window.onerror = ignoreError;

应用于：window 对象

onfocus 事件

发生在窗口得到焦点的时候。应用于：window 对象

onload 事件

发生在文档全部下载完毕的时候。全部下载完毕意味着不但 HTML 文件，而且包含的图片、插件、控件、小程序等全部内容都下载完毕。本事件是 window 的事件，但是在 HTML 中指定事件处理程序的时候，我们是把它写在<body>标记中的。

应用于：window 对象

在 *body* 中使用 *onload* 事件

代码：

```
<html>
  <head>
    <script type="text/javascript">
      function message()
      {
        alert("使用 onload 事件!")
      }
    </script>
  </head>
  <body onload="message()">
  </body>
```

</html> 我想看一

[下:jsCode\userOnload.HTML](#)

onmousedown 事件

发生在用户把鼠标放在对象上按下鼠标键的时候。参考 onmouseup 事件。

应用于：Button 对象；Link 对象

onmouseout 事件

发生在鼠标离开对象的时候。参考 onmouseover 事件。

应用于：Link 对象

onmouseover 事件

发生在鼠标进入对象范围的时候。这个事件和 onmouseout 事件，再加上图片的预读，就可以做到当鼠标移到图像连接上，图像更改的效果了。有时我们看到，在指向一个连接时，状态栏上不显示地址，而显示其它的资料，看起来这些资料是可以随时更改的。它们是这样做出来的：

```
<a href="..."
```

```
onmouseover="window.status='Click Me Please!'; return true;"  
onmouseout="window.status=""; return true;">
```

应用于：Link 对象

onmouseup 事件 发生在用户把鼠标放在对象上鼠标键被按下的情况下，放开鼠标键的时候。如果按下鼠标键的时候，鼠标并不在放开鼠标的对象上，则本事件不会发生。

应用于：Button 对象；Link 对象

onreset 事件 发生在表单的“重置”按钮被单击（按下并放开）的时候。通过在事件处理程序中返回

false 值（return false）可以阻止表单重置。

应用于：Form 对象

onresize 事件 发生在窗口被调整大小的时候。应用于：window 对象

onsubmit 事件 发生在表单的“提交”按钮被单击（按下并放开）的时候。可以使用该事件来验证表单

的有效性。通过在事件处理程序中返回 false 值（return false）可以阻止表单提交。

应用于：Form 对象

onunload 事件

发生在用户退出文档（或者关闭窗口，或者到另一个页面去）的时候。与 onload 一样，要写在 HTML 中就写到<body>标记里。

有的人用这个方法弹出“调查表单”，以“强迫”来者填写；有的就弹出广告窗口，唆使来者点击连接。这种使用“onunload=open...”的方法很不好，有时甚至会因为弹出太多窗口而导致资源缺乏。在用户想离开的时候又打开其他内容是很讨厌的行为。

应用于：window 对象。

第十九讲 基于对象的 JavaScript 语言

JavaScript 语言是基于对象的（Object-Based），而不是面向对象的（object-oriented）。所以说它是一门基于对象的语言，主要是因为它没有提供象抽象、继承、重载等有关面向对象语言的许多功能。而是把其它语言所创建的复杂对象统一起来，从而形成一个非常强大的对象系统。

虽然 JavaScript 语言是一门基于对象的，但它还是具有一些面向对象的基本特征。它可

以根据需要创建自己的对象，从而进一步扩大 JavaScript 的应用范围，增强编写功能强大的 Web 文档。我们在这一讲主要讲

述: **JavaScript** 对象的定义和引用

JavaScript 的常用对象(字符串,数组,日期等)

19.1 Javascript 对象的基础知识

z JavaScript 对象的基本结构

JavaScript 中的对象是由属性(properties)和方法(methods)两个基本的元素构成的。前者是对象在实施其所需要行为的过程中，实现信息的装载单位，从而与变量相关联；后者是指对象能够按照设计者的意图而被执行，从而与特定的函数相联。

z 引用对象的途径

引用 JavaScript 内部对象 由
浏览器环境中提供的对象 创
建新对象

这就是说一个对象在被引用之前，这个对象必须存在，否则引用将毫无意义，而出现错误信息。从上面中我们可以看出 JavaScript 引用对象可通过三种方式获取。要么创建新的对象，要么利用现存的对象。

z 有关对象操作语句

JavaScript 不是一种纯面向对象的语言，但它提供了面向对象语言的一些功能，因此 JavaScript 设计者把它称为“基于对象”而不是面向对象的语言，在 JavaScript 中提供了几个用于操作对象的语句和关键字及运算符。

A for...in 语句

格式如下：

for (对象属性名 in 已知对象名)

说明：

该语句的功能是用于对已知对象的所有属性进行操作的循环。它是将一个已知对象的所有属性反复置给一个变量；而不是使用计数器来实现的,该语句的优点是无需知道对象中属性的个数即可进行操作。

例：下列函数是显示数组中的内容：

```
function showData(object)
```

```
for (var i=0; i<30;i++) document.write(object[i])
```

该函数是通过数组下标顺序值，来访问每个对象的属性，使用这种方式首先必须知道数组的下标值，否则若超出范围，则就会发生错误。

如使用 for...in 语句，则根本不需要知道对象属性的个数，如下：

```
function showData(object){
    for(var prop in object)
        document.write(object[prop]);
}
```

使用该函数时，在循环体中，for 自动将的属性取出来，直到最后为此。

B with 语句 使用该语句的意思是：在该语句体内，任何对变量的引用被认为是这个对象的属性，以

节省一些代码。

```
with object{
```

```
...
```

```
}
```

所有在 with 语句后的花括号中的语句，都是在后面 object 对象的作用域的。

C this 关键字

this 是对当前的引用，在 JavaScript 由于对象的引用是多层次，多方位的，往往一个对象的引用又需要对另一个对象的引用，而另一个对象有可能又要引用另一个对象，这样有可能造成混乱，最后自己已不知道现在引用的那一个对象，为此 JavaScript 提供了一个用于将对象指定当前对象的语句 this。

D New 运算符

虽然在 JavaScript 中对象的功能已经是非常强大的了。但更强大的是设计人员可以按照需求来创建自己的对象，以满足某一特定的要求。使用 New 运算符可以创建一个新的对象。其创建对象使用如下格式：

```
Newobject=NEW Object(Parameters table);
```

其中 Newobject 创建的新对象：object 是已经存在的对象；parameters table 参数表；new 是 JavaScript 中的命令语句。

如创建一个日期新对象

```
newData=New Data()
```

```
birthday=New Data (December 12.1998)
```

之后就可使 newData、birthday 作为一个新的日期对象了。

z 对象属性的引用 对象属性的引用可由下列三种

方式之一实现：

A 使用点 (.) 运算符

```
university.name= "云南省"
```

```
university.city= "昆明市"
```

```
university.date="1999"
```

其中 university 是一个已经存在的对象，name、city、date 是它的三个属性，并通过操作对其赋值。

B 通过对象的下标实现引用

```
university[0]= "云南"
```

```
university[1]= "昆明市"
```

```
university[2]="1999"
```

通过数组形式的访问属性，可以使用循环操作获取其值。

```
function showunievsvity(object){
    for (var j=0;j<2; j++)
        document.write(object[j])
}
```

若采用 For...in 则可以不知其属性的个数后就可以实现：

```
function showmy(object){
    for (var prop in this)
        docament.write(this[prop]);
}
```

C 通过字符串的形式实现

```

university["name"]= "云南"
university["city"]= "昆明市"
university["date"]="1999"

```

z 对象的方法的引用

在 JavaScript 中对象方法的引用是非常简单的。

```
ObjectName.methods()
```

如引用 university 对象中的 showmy () 方法，则可使用：

```
document.write (university.showmy())
```

如引用 math 内部对象中 cos()的方法

则使用：

```
with(Math)
```

```
document.write(cos(35));
```

```
document.write(cos(80));
```

若不使用 with 则引用时相对要复杂些：

```
document.write(Math.cos(35))
```

```
document.write(Math.sin(80))
```

19.2 JavaScript 的常用对象

z String 字符串对象。声明一个字符串对象最简单,常用的方法就是直接赋值。

属性: length 用法: <字符串对象>.length ; 返回该字符串的长度。检测字符串长度的例子

代码:

```

<html>
  <body>
    <script type="text/javascript">
      var str="你好!"
      document.write("<p>" + str + "</p>");
      document.write("字符串长度为:");
      document.write(str.length);
      str="hello boy!"
      document.write("<p>" + str + "</p>");
      document.write("字符串长度为:");
      document.write(str.length)
    </script>
  </body>
</html> 我想看一

```

下:[jsCode\strlenlength.HTML](#) 方法:

charAt()

用法：

`<字符串对象>.charAt(<位置>)`；返回该字符串位于第<位置>位的单个字符。

注意：

字符串中的第一个字符是第 0 位的，第二个才是第 1 位的，最后一个字符是第 `length - 1` 位的。

charCodeAt()

用法：

`<字符串对象>.charCodeAt(<位置>)`；返回该字符串位于第<位置>位的单个字符的 ASCII 码。

indexOf()

用法：

`<字符串对象>.indexOf(<另一个字符串对象>[, <起始位置>])`；该方法从<字符串对象>中查找<另一个字符串对象>（如果给出<起始位置>就忽略之前的位置），如果找到了，就返回它的位置，没有找到就返回“-1”。所有的“位置”都是从零开始的。检测子字符串位置的例子

代码：

```
<html>
  <body>
    <script type="text/javascript">
      var str="SZNSSchools is great!"
      var pos=str.indexOf("School")
      if (pos>=0)
      {
        document.write("发现'School'在 ")
        document.write("'+str+'的第")
        document.write(pos + "个位置!"+"<br />")
      }
      else
      {
        document.write("未发现'school'!")
      }
    </script>
    <p>使用 indexOf 的例子,位置是从 0 开始计算的.</p>
  </body>
</html>
```

我想看一下:[jsCode\stringIndexof.HTML](#)

lastIndexOf()

用法：

`<字符串对象>.lastIndexOf(<另一个字符串对象>[, <起始位置>])`；跟 `indexOf()` 相似，不过是从后边开始找。

split()

用法：

`<字符串对象>.split(<分隔符字符>)`；返回一个数组，该数组是从<字符串对象>中分离出来的，<分隔符字符>决定了分离的地方，它本身不会包含在所返回的数组中。

例如：`'1&2&345&678'.split('&')`返回数组：`1,2,345,678`。关于数组，我们将在下面讨论。

substring()

用法：

<字符串对象>.substring(<始>[, <终>]); 返回原字符串的子字符串, 该字符串是原字符串从<始>位置到<终>位置的前一位置的一段。<终> - <始> = 返回字符串的长度 (length)。如果没有指定<终>或指定得超过字符串长度, 则子字符串从<始>位置一直取到原字符串尾。如果所指定的位置不能返回字符串, 则返回空字符串。

substr()

用法：

<字符串对象>.substr(<始>[, <长>]); 返回原字符串的子字符串, 该字符串是原字符串从<始>位置开始, 长度为<长>的一段。如果没有指定<长>或指定得超过字符串长度, 则子字符串从<始>位置一直取到原字符串尾。如果所指定的位置不能返回字符串, 则返回空字符串。

截取子字符串的例子

代码:

<html>

<body>

```

    <script type="text/javascript"> var
    str="SZNSSchools is great!"
    document.write(str.substr(2,6))
    document.write("<br /><br />")
    document.write(str.substring(2,6))
  </script>

```

<p>

substr()返回字符串的特定部分. 第一个参数是截取的位置(第一个位置为 0), 第二个参数是截取的长度.

</p>

<p>

substring()返回字符串的特定部分. 第一个参数是截取的位置(第一个位置为 0), 第二个参数是结束的位置.

</p>

</body>

</html>

我想看一下:[jsCode\strsubstr.HTML](#)***toLowerCase()***

用法：

<字符串对象>.toLowerCase(); 返回把原字符串所有大写字母都变成小写的字符串。

toUpperCase()

用法：

<字符串对象>.toUpperCase(); 返回把原字符串所有小写字母都变成大写的字符串。

转换字符串的大小写

代码:

```

<html>
<body>
  <script type="text/javascript"> var
  str=("Hello JavaScripters!")
  document.write(str.toLowerCase())
  document.write("<br>")
  document.write(str.toUpperCase())
  </script>
</body>
</html>

```

我想看一下:[jsCode\strUpperLower.HTML](#)

match()

用法:

<字符串对象>.match(); 在源字符串中查找指定子串,如存在返回指定字符串,如不存在返回 null. 检测

子串代

码:

```

<html>
  <body>
    <script type="text/javascript">
      var str = "SZNSSchools is great!";
      document.write(str.match("great"));
      document.write("<br>");
      str = "SZNSSchools is ggreat!";
      document.write(str.match("great"));
      document.write("<br>");
      str = "SZNSSchools is ggreatt!";
      document.write(str.match("great"));
      document.write("<br>");
      str = "SZNSSchools is grreat!";
      document.write(str.match("great"));
    </script>
    <p>检测单词是否存在,如存在返回该单词,如不存在,返回 null</p>
  </body>
</html> 我想看一

```

下:[jsCode\strMatch.HTML](#) z

Array 数组对象

数组对象是一个对象的集合,里边的对象可以是不同类型的(这和 **Java** 是不同的).数组的每一个成员对象都有一个“下标”,用来表示它在数组中的位置(从 0 开始)。数组的定义方法:

```
var <数组名> = new Array(size);
```

这样就定义了一个空数组。以后要添加数组元素,就用:

```
<数组名>[<下标>] = ...
```

比如:

```
var myArr=new Array(3);
myArr[0]=1;myArr[1]=2;myArr[2]=3;
```

如果想在定义数组的时候直接初始化数据可以用：

```
var <数组名> = new Array(<元素 1>, <元素 2>, <元素 3>...);
```

例如

```
var myArray = new Array(1, 4.5, 'Hi'); 定义了一个数组 myArray，里边的元素是：myArray[0]
=1; myArray[1] = 4.5; myArray[2] = 'Hi'。但是，如果元素列表中只有一个元素，而这个元素
又是一个正整数的话，比如，var myArr=new Array(3)表示包含 3 个元素的数组，使用 new 定义
数组时，使用的是0而不是[]。
```

注意：JavaScript 只有一维数组！二维数组是使用一维数组中定义一维数组等变通的方式来实现。

属性

length

用法：

<数组对象>.length；返回：数组的长度，即数组里有多少个元素。它等于数组里最后一个元素的下标加一。

方法

join()

用法：

<数组对象>.join(<分隔符>)；返回一个字符串，该字符串把数组中的各个元素串起来，用<分隔符>置于元素与元素之间，如果不指定分隔符默认用“,”连接。这个方法不影响数组原本的内容。

reverse()

用法：

<数组对象>.reverse()；使数组中的元素顺序反过来。如果对数组[1, 2, 3]使用这个方法，它将使数组变成：[3, 2, 1]。

slice()

用法：

<数组对象>.slice(<始>[, <终>])；返回一个数组，该数组是原数组的子集，始于<始>，终于<终>。如果不给出<终>，则子集一直取到原数组的结尾。

sort()

用法：

<数组对象>.sort([<方法函数>])；使数组中的元素按照一定的顺序排列。如果不指定<方法函数>，则按字母顺序排列。在这种情况下，80 是比 9 排得前的。如果指定<方法函数>，则按<方法函数>所指定的排序方法排序。<方法函数>比较难讲述，这里只将一些有用的<方法函数>介绍给大家。

按升序排列数字：

```
function sortMethod(a, b) {
    return a - b;
```

```
} myArray.sort(sortMethod); 按降序排列数字：把上
```

面的“a - b”该成“b - a”。最简单的数组应用

代码:

```
<html>
  <body>
    <script type="text/javascript">
      var famname = new Array(6)
      famname[0] = "Jan Egil"
      famname[1] = "Tove"
      famname[2] = "Hege"
      famname[3] = "Stale"
      famname[4] = "Kai Jim"
      famname[5] = "Borge"

      for (i=0; i<6; i++)
      {
        document.write(famname[i] + "<br>")
      }
    </script>
  </body>
</html>
```

我想看一下:[jsCode\arrSample.HTML](#)

声明数组时直接赋值

代码:

```
<html>
  <body>
    <script type="text/javascript">
      var famname = new Array("Jan Egil","Tove","Hege","Stale","Kai Jim","Borge")

      for (i=0; i<famname.length; i++)
      {
        document.write(famname[i] + "<br>")
      }
    </script>
  </body>
</html> 我想看一
```

下:[jsCode\arrSample2.HTML](#) 使用数

组的一些属性和方法

```
<html>
  <body>
    <script type="text/javascript">
      var famname = new Array(3)
      famname[0] = "Jani"
      famname[1] = "Tove"
      famname[2] = "Hege"
      document.write("字符串长度:");
```

```

document.write(famname.length + "<br>")
document.write("使用.链接成一个字符串:<br>");
document.write(famname.join(".") + "<br>") document.write("使数组
中的元素反序排列:<br>"); document.write(famname.reverse() +
"<br>") document.write("在使数组中的增加两个元素,返回数组长
度:<br>");
document.write(famname.push("Ola","Jon") + "<br>")
document.write(famname.join(".") + "<br>")
document.write("取出最后一个元素,数组长度减少:<br>");
document.write(famname.pop() + "<br>")
document.write(famname.join(".") + "<br>")
document.write("取出开始的元素,数组长度减少:<br>");
document.write(famname.shift() + "<br>")
document.write(famname.join(".") + "<br>")
document.write(famname.shift() + "<br>")
document.write(famname.join(".") + "<br>")

```

```
</script>
```

```
</body>
```

```
</html> 我想看一
```

下:[jsCode\strUseage.HTML](#) 使用数组
的两个方法 *concat* 和 *slice* 代码:

```
<html>
```

```
<body>
```

```

<script type="text/javascript">
    var famname = new Array(3)
    famname[0] = "Jani"
    famname[1] = "Tove"
    famname[2] = "Hege"

    var famname2 = new Array(3)
    famname2[0] = "John"
    famname2[1] = "Andy"
    famname2[2] = "Wendy"

    var famname3 = new Array("Stale", "Borge")
    document.write("使用 join()连接<br>");
    document.write(famname.join() + "<br>")
    document.write("concat(famname2):<br>");
    document.write(famname.concat(famname2) + "<br>")
    document.write("famname 没有变化:<br>");
    document.write(famname.join() + "<br>")
    document.write("concat(famname2,famname3):<br>");
    document.write(famname.concat(famname2,famname3) + "<br>")

```

```

document.write("famname 没有变化:<br>");
document.write(famname.join() + "<br>")
document.write("slice(1):<br>");
document.write(famname.slice(1) + "<br>")

```

```
</script>
```

```
</body>
```

</html>使用 concat 返回的数组相加后的数组,并不会改变源数组,slice 从给定的位置截取数组,不指定结束位置则截到结尾,不会影响源数组.我想看一

下:[jsCode\strConcatSlice.HTML](#)

z Math “数学” 对象 提供对数据的数学计算。下面所提到的属性和方法，不再详细说明“用法”，大家在使

用的时候记住用 “Math.<名>” 这种格式。

属性

E 返回常数 e (2.718281828...)。LN2

返回 2 的自然对数 (ln 2)。LN10 返

回 10 的自然对数 (ln 10)。

LOG2E 返回以 2 为底的 e 的对数 (log₂e)。

LOG10E 返回以 10 为底的 e 的对数 (log₁₀e)。

PI 返回π (3.1415926535...)。

SQRT1_2 返回 1/2 的平方根。

SQRT2 返回 2 的平方根。

方法

abs(x) 返回 x 的绝对值。

acos(x) 返回 x 的反余弦值 (余弦值等于 x 的角度) , 用弧度表示。

asin(x) 返回 x 的反正弦值。

atan(x) 返回 x 的反正切值。

atan2(x, y) 返回复平面内点(x, y)对应的复数的幅角, 用弧度表示, 其值在 $-\pi$ 到 π 之间。

ceil(x) 返回大于等于 x 的最小整数。

cos(x) 返回 x 的余弦。

exp(x) 返回 e 的 x 次幂 (ex)。

floor(x) 返回小于等于 x 的最大整数。

log(x) 返回 x 的自然对数 (ln x)。

max(a, b) 返回 a, b 中较大的数。

min(a, b) 返回 a, b 中较小的数。

pow(n, m) 返回 n 的 m 次幂 (nm)。

random() 返回大于 0 小于 1 的一个随机数。

round(x) 返回 x 四舍五入后的值。

sin(x) 返回 x 的正弦。

sqrt(x) 返回 x 的平方根。

tan(x) 返回 x 的正切。用

随机数产生链接的例子 代

码:

```
<html>
```

```
<body>
```

下面的网址是随机产生的!


```
<script type="text/javascript">
    var r=Math.random()
    if (r>0.5)
    {
        document.write("<a href='http://www.cjol.com'>中国人才热线!</a>")
    }
    else
    {
        document.write("<a href='http://www.51job.com'>前程无忧!</a>")
    }
</script>
</body>
```

</html> 我想看一

下:[jsCode\randomlink.HTML](#) 使用

Math 综合例子

<html>

<body>

```
<script type="text/javascript">
    document.write("Math.round(7.56)=");
    document.write(Math.round(7.56)+"<br>");
    document.write("产生 0-1 之间的随机数:");
    document.write(Math.random()+"<br>");
    document.write("产生 0-10 之间的随机数:");
    document.write(Math.round(Math.random()*10)+"<br>");
    document.write("求最大值:");
    document.write(Math.max(2,40)+"<br>");
    document.write("求最小值:");
    document.write(Math.min(2,40)+"<br>")
</script>
```

</body>

</html> 我想看一

下:[jsCode\math.HTML](#) z

Date 日期对象

这个对象可以储存任意一个日期，从 0001 年到 9999 年，并且可以精确到毫秒数 (1/1000 秒)。

在内部，日期对象是一个整数，它是从 1970 年 1 月 1 日零时整开始计算到日期对象所指的日期的毫秒数。如果所指日期比 1970 年早，则它是一个负数。

所有日期时间，如果不指定时区，都采用 "UTC" (世界时) 时区，它与 "GMT" (格林威治时间) 在数值上是一样的。

定义一个日期对象：

```
var d = new Date;
```

这个方法使 d 成为日期对象，并且已有初始值：当前时间。如果要自定初始值，可以用：

```
var d = new Date(99, 10, 1); //99 年 10 月 1 日
var d = new Date('Oct 1, 1999'); //99 年 10 月 1 日
```

等等方法。

最好的方法就是用下面介绍的“方法”来严格的定义时间。

方法说明:以下有很多“*g/set[UTC]XXX*”这样的方法,它表示既有“*getXXX*”方法,又有“*setXXX*”方法。“*get*”是获得某个数值,而“*set*”是设定某个数值。如果带有“*UTC*”字母(如 *getUTCXXX*),则表示获得/设定的数值是基于 *UTC* 时间的,没有则表示基于本地时间或浏览默认时间。如无说明,方法的使用格式为:“<对象>.<方法>”,下同。

g/set[UTC]FullYear()

返回/设置年份,用四位数表示。如果使用“*x.set[UTC]FullYear(99)*”,则年份被设定为 0099 年,*x.setFullYear(2000)*表示 2000 年。

g/set[UTC]Year()

返回/设置年份,用两位数表示。设定的时候浏览器自动加上“19”开头,故使用“*x.set[UTC]Year(00)*”把年份设定为 1900 年。2000 年以后的时间要用 *set/getFullYear()*。

g/set[UTC]Month()

返回/设置月份,取值 0-11,0 表示一月,和习惯不同要注意,使用时要注意加 1。

g/set[UTC]Date()

返回/设置日期,取值 1-31,不是从 0 开始的,JavaScript 很少从 1 开始计算。使用时要注意。

get[UTC]Day()

返回/设置星期,取值 0-6,0 表示星期天,6 表示星期六,没有对应的 *set* 方法因为星期是自动计算的,不用设置。

g/set[UTC]Hours()

返回/设置小时数,取值 0-23,0 为午夜 24 点,24 小时制。

g/set[UTC]Minutes()

返回/设置分钟数,取值 0-59。

g/set[UTC]Seconds()

返回/设置秒钟数,取值 0-59。

g/set[UTC]Milliseconds()

返回/设置毫秒数,取值 0-999。

g/setTime()

返回/设置时间,该时间就是日期对象的内部处理方法,用毫秒表示,从 1970 年 1 月 1 日零时正开始计算到日期对象所指的日期的毫秒数。如果要使某日期对象所指的时间推迟 1 小时,就用:“*x.setTime(x.getTime() + 60 * 60 * 1000);*”(一小时 60 分,一分 60 秒,一秒 1000 毫秒)。

getTimezoneOffset()

返回日期对象采用的时区与格林威治时间所差的分钟数。在格林威治东方的市区,该值为负,例如:中国时区(GMT+0800)返回“-480”。

toString()

返回一个字符串,描述日期对象所指的日期。这个字符串的格式类似于:“Fri Jul 21 15:43:46 UTC+0800 2000”。

toLocaleString()

返回一个字符串,描述日期对象所指的日期,用本地时间表示格式。如:“2000-07-21 15:43:46”。

toGMTString()

返回一个字符串，描述日期对象所指的日期，用 GMT 格式。

toUTCString()

返回一个字符串，描述日期对象所指的日期，用 UTC 格式。

parse()

用法：Date.parse(<日期对象>),Date.parse("Jan 11,1996")将字符串代表的时间转换成毫秒 (Date 内部的日期表示)。

19.3 综合例子

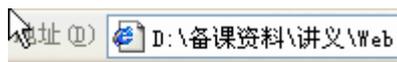
显示今天的日期

代码:

```
<html>
  <body>
    <script type="text/javascript">
      var d = new Date()
      document.write("今天是:");
      document.write(d.getFullYear())
      document.write("-")
      document.write(d.getMonth() + 1)
      document.write("-")
      document.write(d.getDate())
    </script>
  </body>
</html>
```

注意:月份要加 **1**,默认从 **0** 开始.

我想看一下:[jsCode\dateToday.HTML](#)



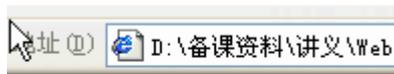
今天是:2005-8-26

显示当前时间

代码:

```
<html>
  <body>
    <script type="text/javascript">
      var d = new Date()
      document.write("当前时间是:");
      document.write(d.getHours())
      document.write(".")
      document.write(d.getMinutes())
      document.write(".")
      document.write(d.getSeconds())
    </script>
```

```
</body>
</html>
我想看一下:jsCode\dateCurrentTime.HTML
```



当前时间是:22. 38. 6

设置日期

代码:

```
<html>
  <body>
    <script type="text/javascript">
      var d = new Date()
      document.write("设定日期:")
      d.setFullYear("1990")
      document.write(d)
    </script>
  </body>
</html>
```

我想看一

下:[jsCode\setyear.HTML](#) 使用

UTC 时间:

代码:

```
<html>
  <body>
    <script type="text/javascript"> var d
    = new Date() document.write("UTC
    时间:");
    document.write(d.getUTCHours())
    document.write(".")
    document.write(d.getUTCMinutes())
    document.write(".")
    document.write(d.getUTCSeconds())
    </script>
  </body>
</html>
```

我们发现和我们本地时间相差 8 小时:

我想看一下:[jsCode\utcTime.HTML](#) 显

示当前星期

代码:

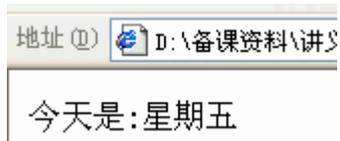
```
<html>
  <body>
    <script type="text/javascript">
      var d=new Date()
      var weekday=new Array("星期日","星期一","星期二","星期三","星期四","星期五","
```

```

星期六")
    document.write("今天是:" + weekday[d.getDay()])
    </script>
</body>
</html>

```

我想看一下:[jsCode\week.HTML](#)



显示当前的日期和星期

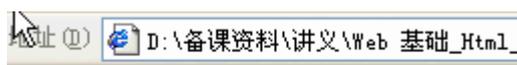
代码:

```

<html>
  <body>
    <script type="text/javascript">
      var d=new Date()
      var weekday=new Array("星期日","星期一","星期二","星期三","星期四","星期
五","星期六")
      var monthname=new Array("一月","二月","三月","四月","五月","六月","七月","
八月","九月","十月","十一月","十二月") document.write("今
      天是:"); document.write(d.getFullYear()+"年")
      document.write(monthname[d.getMonth()] + " ")
      document.write(d.getDate() + "日 ")
      document.write(weekday[d.getDay()] + " ")
    </script>
  </body>
</html>

```

我想看一下:[jsCode\monthandweek.HTML](#)



今天是:2005年八月 26日 星期五

一个走动的时间

代码:

```

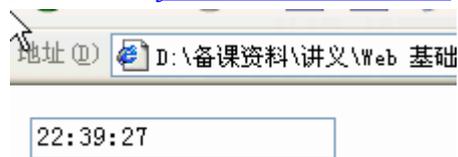
<html>
  <head>
    <script type="text/javascript">
      var timer = null

      function stop()
      {
        clearTimeout(timer)
      }
    </script>
  </head>

```

```
function start()
{
    var time = new Date()
    var hours = time.getHours()
    var minutes = time.getMinutes()
    minutes=((minutes < 10) ? "0" : "") + minutes
    var seconds = time.getSeconds()
    seconds=((seconds < 10) ? "0" : "") + seconds
    var clock = hours + ":" + minutes + ":" + seconds
    document.forms[0].display.value = clock
    timer = setTimeout("start()",1000)
}
</script>
</head>
<body onload="start()" onunload="stop()">
    <form>
        <input type="text" name="display" size="20">
    </form>
</body>
</html>
```

我想看一下:[jsCode\moveTime.HTML](#)



第二十讲 系统函数,常用的内部对象

我们在这一讲的主要内容是:

系统函数 常用的内部对象

20.1 系统函数

JavaScript 中的系统函数又称内部方法。它提供了与任何对象无关的系统函数，使用这

些函数不需创建任何实例,可直接用。

eval()

eval (字符串表达式)

返回字符串表达式中的值,例如: test=eval("8+9+5/2");

isNaN()

如果括号内的值是 "NaN" 则返回 true 否则返回 false。

parseInt()

返回把括号内的内容转换成整数之后的值。如果括号内是字符串,则字符串开头的数字部分被转换成整数,如果以字母开头,则返回 "NaN"。

parseFloat()

返回把括号内的字符串转换成浮点数之后的值,字符串开头的数字部分被转换成浮点数,如果以字母开头,则返回 "NaN"。

toString() 用法: <对象>.toString(); 把对象转换成字符串。如果在括号中指定一个数值,则转换过程中所有数值转换成特定进制。

escape()

返回括号中的字符串经过编码后的新字符串。该编码应用于 URL,也就是把空格写成 "%20" 这种格式。"+" 不被编码,如果要 "+" 也被编码,请用: escape('...', 1)。

unescape()

是 escape() 的反过程。解编括号中字符串成为一般字符串。

20.2 常用的内部对象

使用浏览器的内部对象系统,可实现与 HTML 文档进行交互。它的作用是将相关元素组织包装起来,提供给程序设计人员使用,从而减轻编程人的劳动,提高设计 Web 页面的能力。

20.2.1 内部对象树

• navigator	浏览器对象
• screen	屏幕对象
• window	窗口对象
○ history	历史对象
○ location	地址对象
○ frames[]: Frame	框架对象
○ document	文档对象
■ anchors[]; links[]; Link	连接对象
■ applets[]	Java小程序对象
■ embeds[]	插件对象
■ forms[]: Form	表单对象
■ Button	按钮对象
■ Checkbox	复选框对象
■ elements[]: Element	表单元素对象
■ Hidden	隐藏对象
■ Password	密码输入区对象
■ Radio	单选域对象
■ Reset	重置按钮对象
■ Select	选择区(下拉菜单、列表)对象
■ options[]: Option	选择项对象
■ Submit	提交按钮对象
■ Text	文本框对象
■ Textarea	多行文本输入区对象
■ images[]: Image	图片对象

20.2.2 navigator 浏览器对象

反映了当前使用的浏览器的资料。

属性

appCodeName

返回浏览器的“码名”，流行的 IE 返回 'Mozilla'。

appName

返回浏览器名。IE 返回 'Microsoft Internet Explorer'

appVersion

返回浏览器版本，包括了大版本号、小版本号、语言、操作平台等信息。

platform

返回浏览器的操作平台，对于 Windows 9x 上的浏览器，返回 'Win32'（大小写可能有差异）。

userAgent

返回以上全部信息。例如，IE5.01 返回 'Mozilla/4.0 (compatible; MSIE 5.01; Windows 98)'。

显示浏览器详细的信息

```
<html>
  <body>
    <script type="text/javascript">
      document.write("<p>浏览器: ")
      document.write(navigator.appName + "</p>")

      document.write("<p>版本: ")
      document.write(navigator.appVersion + "</p>")

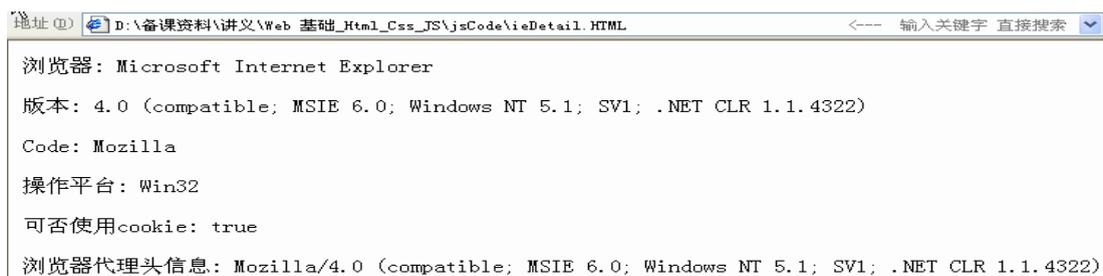
      document.write("<p>Code: ")
      document.write(navigator.appCodeName + "</p>")

      document.write("<p>操作平台: ")
      document.write(navigator.platform + "</p>")

      document.write("<p>可否使用 cookie: ")
      document.write(navigator.cookieEnabled + "</p>")

      document.write("<p>浏览器代理头信息: ")
      document.write(navigator.userAgent + "</p>")
    </script>
  </body>
</html>
```

我想看一下:jsCode\ieDetail.HTML



20.2.3 screen 屏幕对象

反映了当前用户的屏幕设置。

属性

width

返回屏幕的宽度 (像素数)。

height

返回屏幕的高度。

availWidth

返回屏幕的可用宽度 (除去了一些不自动隐藏的类似任务栏的东西所占用的宽度)。

availHeight

返回屏幕的可用高度。

colorDepth

返回当前颜色设置所用的位数 - 1 : 黑白 ; 8 : 256 色 ; 16 : 增强色 ; 24/32 : 真彩色

20.2.4 window 窗口对象

最大的对象，它描述的是一个浏览器窗口。一般要引用它的属性和方法时，不需要用“window.xxx”这种形式，而直接使用“xxx”。一个框架页面也是一个窗口。

属性

name

窗口的名称，由打开它的连接 () 或框架页 (<frame name="...">) 或某一个窗口调用的 open() 方法 (见下) 决定。一般我们不会用这个属性。

status

指窗口下方的“状态栏”所显示的内容。通过对 status 赋值，可以改变状态栏的显示。

opener

用法 window.opener 返回打开

本窗口的窗口对象。

注意：返回的是一个窗口对象。如果窗口不是由其他窗口打开的，在 **Netscape** 中这个属性返回 **null**；在 **IE** 中返回“未定义” (**undefined**)。undefined 在一定程度上等于 **null**。注意：**undefined** 不是 **JavaScript** 常数。

self

指窗口本身，它返回的对象跟 window 对象是一模一样的。最常用的是“self.close()”，放在<a>标记中：“关闭窗口”。

parent

返回窗口所属的框架页对象。

top

返回占据整个浏览器窗口的最顶端的框架页对象。

方法**open()**

打开一个窗口。

用法

open(<URL 字符串>, <窗口名称字符串>, <参数字符串>);

<URL 字符串>: 描述所打开的窗口打开哪一个网页。如果留空(""), 则不打开任意网页。

<窗口名称字符串>: 描述被打开的窗口的名称 (window.name), 可以使用'_top'、'_blank' 等内建名称。这里的名称跟 "" 里的 "target" 属性是一样的。

<参数字符串>: 描述被打开的窗口的样貌。如果只需要打开一个普通窗口, 该字符串留空(""), 如果要指定样貌, 就在字符串里写上一到多个参数, 参数之间用逗号隔开。例如: 打开一个 400 x 100 的干净的窗口:

```
open('_', '_blank', 'width=400,height=100,menubar=no,toolbar=no,
location=no,directories=no,status=no,scrollbars=yes,resizable=yes')
```

参数

top=# 窗口顶部离开屏幕顶部的像素数

left=# 窗口左端离开屏幕左端的像素数

width=# 窗口的宽度

height=# 窗口的高度

menubar=... 窗口有没有菜单, 取值 yes 或 no

toolbar=... 窗口有没有工具条, 取值 yes 或 no

location=... 窗口有没有地址栏, 取值 yes 或 no

directories=... 窗口有没有连接区, 取值 yes 或 no

scrollbars=... 窗口有没有滚动条, 取值 yes 或 no

status=... 窗口有没有状态栏, 取值 yes 或 no

resizable=... 窗口给不给调整大小, 取值 yes 或 no

open() 方法有返回值, 返回的就是它打开的窗口对象。所以, var newWindow = open('_', '_blank'); 这样把一个新窗口赋值到 "newWindow" 变量中, 以后通过 "newWindow" 变量就可以控制窗口了。

close()

关闭一个已打开的窗口。

用法

window.close() 或 self.close(): 关闭本窗口;

<窗口对象>.close(): 关闭指定的窗口。如果该窗口有状态栏, 调用该方法后浏览器会警告: "网页正在试图关闭窗口, 是否关闭?" 然后等待用户选择是否; 如果没有状态栏, 调用该方法将直接关闭窗口。

blur()

使焦点从窗口移走, 窗口变为 "非活动窗口"。

focus()

使窗口获得焦点，变为“活动窗口”。

scrollTo()

用法

[<窗口对象>].scrollTo(x, y); 使窗口滚动，使文档从左上角数起的(x, y)点滚动到窗口的左上角。

scrollBy()

用法

[<窗口对象>].scrollBy(deltaX, deltaY); 使窗口向右滚动 deltaX 像素，向下滚动 deltaY 像素。如果取负值，则向相反的方向滚动。

resizeTo()

用法

[<窗口对象>].resizeTo(width, height); 使窗口调整大小到宽 width 像素，高 height 像素。

resizeBy()

用法

[<窗口对象>].resizeBy(deltaWidth, deltaHeight); 使窗口调整大小，宽增大 deltaWidth 像素，高增大 deltaHeight 像素。如果取负值，则减少。

alert()

用法

alert(<字符串>); 弹出一个只包含“确定”按钮的对话框，显示<字符串>的内容，整个文档的读取、Script 的运行都会暂停，直到用户按下“确定”。

confirm() 用法 confirm(<字符串>); 弹出一个包含“确定”和“取消”按钮的对话框，显示<字符串>的内容，要求用户做出选择，整个文档的读取、Script 的运行都会暂停。如果用户按下“确定”，则返回 true 值，如果按下“取消”，则返回 false 值。

prompt()

用法

prompt(<字符串>[, <初始值>]); 弹出一个包含“确认”“取消”和一个文本框的对话框，显示<字符串>的内容，要求用户在文本框输入一些数据，整个文档的读取、Script 的运行都会暂停。如果用户按下“确认”，则返回文本框里已有的内容，如果用户按下“取消”，则返回 null 值。如果指定<初始值>，则文本框里会有默认值。

事件

onload; onunload; onresize; onblur; onfocus; onerror

20.2.5 history 历史对象

历史对象指浏览器的浏览历史。

属性

length

历史的项数。JavaScript 所能管到的历史被限制在用浏览器的“前进”“后退”键可以去到的范围。本属性返回的是“前进”和“后退”两个按键之下包含的地址数的和。

方法

back()

后退，跟按下“后退”键是等效的。

forward()

前进，跟按下“前进”键是等效的。

go()

用法

history.go(x)；在历史的范围内去到指定的一个地址。如果 $x < 0$ ，则后退 x 个地址，如果 $x > 0$ ，则前进 x 个地址，如果 $x = 0$ ，则刷新现在打开的网页。history.go(0) 跟 location.reload() 是等效的。

20.2.6 location 地址对象

它描述的是某一个窗口对象所打开的地址。要表示当前窗口的地址，只需要使用“location”就行了；若要表示某一个窗口的地址，就使用“<窗口对象>.location”。

属性

protocol

返回地址的协议，取值为 'http:', 'https:', 'file:' 等等。

hostname

返回地址的主机名，例如，一个“http://www.microsoft.com/china/”的地址，location.hostname == 'www.microsoft.com'。

port

返回地址的端口号，一般 http 的端口号是 '80'。

host

返回主机名和端口号，如：'www.a.com:8080'。

pathname

返回路径名，如“http://www.a.com/b/c.html”，location.pathname == 'b/c.html'。

search

返回“?”以及以后的内容，如“http://www.a.com/b/c.asp?selection=3&jumpto=4”，location.search == '?selection=3&jumpto=4'；如果地址里没有“?”，则返回空字符串。

href

返回以上全部内容，也就是说，返回整个地址。在浏览器的地址栏上怎么显示它就怎么返回。如果想一个窗口对象打开某地址，可以使用“location.href = '...’”，也可以直接用“location = '...’”来达到此目的。

方法

reload()

相当于按浏览器上的“刷新” (IE)或“Reload” (Netscape)键。

replace()

打开一个 URL，并取代历史对象中当前位置的地址。用这个方法打开一个 URL 后，按下浏览器的“后退”键将不能返回到刚才的页面。

frames[]

Frame 框架对象(后面讲述)

刷新页面

代码:

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

```

    <script type="text/javascript">
    function refresh()
    {
    window.location.reload()
    }
    </script>
</head>
<body>
    <script type="text/javascript">
    var d = new Date() document.write("当前时间是:");
    document.write(d.getHours())
    document.write(".");
    document.write(d.getMinutes())
    document.write(".");
    document.write(d.getSeconds())
    </script>
    <form>
    <input type="button" value="刷新页面" onclick="refresh()">
    </form>
</body>
</html>

```

我想看一下:[jsCode\reload.HTML](#)

20.2.7 document 文档对象

描述当前窗口或指定窗口对象的文档。它包含了文档从<head>到</body>的内容。

用法

document (当前窗口) 或 <窗口对象>.document (指定窗口)

属性

cookie

后面讲述

lastModified

当前文档的最后修改日期，是一个 Date 对象。

referrer

如果当前文档是通过点击连接打开的，则 referrer 返回原来的 URL。

title

指<head>标记里用<title>...</title>定义的文字。

fgColor

指<body>标记的 text 属性所表示的文本颜色。

bgColor

指<body>标记的 bgcolor 属性所表示的背景颜色。

linkColor

指<body>标记的 link 属性所表示的连接颜色。

alinkColor

指<body>标记的 alink 属性所表示的活动连接颜色。

vlinkColor

指<body>标记的 vlink 属性所表示的已访问连接颜色。

方法

write(),writeln()

向文档写入数据，所写入的会当成标准文档 HTML 来处理。writeln() 与 write() 的不同点在于，writeln() 在写入数据以后会加一个换行。这个换行只是在 HTML 中换行，具体情况能不能够是显示出来的文字换行，要看插入 JavaScript 的位置而定。如在<pre>标记中插入，这个换行也会体现在文档中。

close()

关闭文档，停止写入数据。

20.2.8 anchors[]; links[]; link 连接对象

用法

document.anchors[[x]]; document.links[[x]]; <anchorId>; <linkId>

document.anchors

是一个数组，包含了文档中所有锚标记（包含 name 属性的<a>标记），按照在文档中的次序，从 0 开始给每个锚标记定义了一个下标。

document.links

也是一个数组，包含了文档中所有连接标记（包含 href 属性的<a>标记和<map>标记段里的<area>标记），按照在文档中的次序，从 0 开始给每个连接标记定义了一个下标。

如果一个<a>标记既有 name 属性，又有 href 属性，则它既是一个 Anchor 对象，又是一个 link 对象。

anchors 和 links 作为数组，有数组的属性和方法。

单个 Anchor 对象没有属性；

单个 Link 对象的属性见下。

属性

protocol; hostname; port; host; pathname; hash; search; href 与 location 对象相同。

target 返回/指定连接的目标窗口（字符串），与<a>标记里的 target 属性是一样的。

事件

onclick; onmouseover; onmouseout; onmousedown; onmouseup

20.2.9 embeds[] 插件对象

它是一个数组，包含了文档中所有的插件（<embed>标记）。因为每个插件的不同，每个 Embed 对象也有不同的属性和方法。

20.2.10 forms[]; Form 表单对象

document.forms[] 是一个数组，包含了文档中所有的表单 (<form>)。要引用单个表单，可以用 document.forms[x]，但是一般来说，人们都会这样做：在<form>标记中加上“name=...”属性，那么直接用“document.<表单名>”就可以引用了。

Form 对象的属性

name

返回表单的名称，也就是<form name=“...”>属性。

action

返回/设定表单的提交地址，也就是<form action=“...”>属性。

method

返回/设定表单的提交方法，也就是<form method=“...”>属性。

target

返回/设定表单提交后返回的窗口，也就是<form target=“...”>属性。

encoding

返回/设定表单提交内容的编码方式，也就是<form enctype=“...”>属性。

length

返回该表单所含元素的数目。

方法

reset()

重置表单。这与按下“重置”按钮是一样的。

submit()

提交表单。这与按下“提交”按钮是一样的。

事件

onreset; onsubmit 自己控制

表单的重置和提交 代码:

```
<html>
  <head>
    <script type="text/javascript">
      function formReset()
      {
        var x=document.forms.myForm;
        x.reset();
        alert("表单已经重置");
      }
      function formSubmit()
      {
        var x=document.forms.myForm;
        x.submit(); alert("表单已经提交");
      }
    </script>
```

```

</head>
<body>
  <form name="myForm">
    <p>控制表单的提交和重置</p>
    <input type="text" size="20"><br>
    <input type="text" size="20"><br>
    <br>
    <input type="button" onclick="formReset()" value="重置">
    <input type="button" onclick="formSubmit()" value="提交">
  </form>
</body>
</html> 我想看一下:jsCode\fromResetSubmit.HTML 以下
从 "Button" 到 "Textarea" 都是表单的元素对象:

```

20.2.10.1 Button 按钮对象

由 "`<input type="button">`" 指定。引用一个 Button 对象，可以使用 "`<文档对象>.<表单对象>.<按钮名称>`"。`<按钮名称>`指在`<input>`标记中的 "`name="..."`" 属性的值。引用任意表单元素都可以用这种方法。

属性

name

返回/设定用`<input name="...">`指定的元素名称。

value

返回/设定用`<input value="...">`指定的元素的值。

form

返回包含本元素的表单对象。

方法

blur()

从对象中移走焦点。

focus()

让对象获得焦点。

click()

模拟鼠标点击该对象。

事件

onclick; onmousedown; onmouseup

20.2.10.2 Checkbox 复选框对象

由 "`<input type="checkbox">`" 指定。

属性

name

返回/设定用`<input name="...">`指定的元素名称。

value

返回/设定用<input value="...">指定的元素的值。

form

返回包含本元素的表单对象。

checked

返回/设定该复选框对象是否被选中。这是一个布尔值。

defaultChecked

返回/设定该复选框对象默认是否被选中。这是一个布尔值。

方法

blur()

从对象中移走焦点。

focus()

让对象获得焦点。

click()

模拟鼠标点击该对象。

事件 onclick 复选框

的选择和取消 代码:

```
<html>
  <head>
    <script type="text/javascript">
      function check()
      {
        var x=document.forms.myForm
        x[0].checked=true
      }
      function uncheck()
      {
        var x=document.forms.myForm
        x[0].checked=false
      }
    </script>
  </head>
  <body>
    <form name="myForm">
      <input type="checkbox" value="on">复选框<br>
      <input type="button" onclick="check()" value="选择">
      <input type="button" onclick="uncheck()" value="取消选择">
    </form>
  </body>
</html> 我想看一
```

下:[jsCode\checkBox1.HTML](#) 获得复选框的值

```

<html>
  <head>
    <script type="text/javascript">
      function check()
      {
        className=document.forms[0].className
        classSec=document.forms[0].classSec
        txt=""
        for (i=0;i<className.length;++ i)
        {
          if (className[i].checked)
          {
            txt=txt + className[i].value + " "
          }
        }
        classSec.value="选择培训的内容:" + txt
      }
    </script>
  </head>
  <body>
    <form>
      选择你要培训的课程<br><br>
      <input type="checkbox" name="className" value="J2se">J2se 课程<br>
      <input type="checkbox" name="className" value="J2ee">J2ee 课程<br><br>
      <input type="button" name="test" onclick="check()" value="我要培训">
      <br><br>
      <input type="text" name="classSec" size="50">
    </form>
  </body>
</html>

```

我想看一下:[jsCode\chockbox2.HTML](#)

地址 (D)  D:\备课资料\讲义\Web 基础_Html_Css_JS\jsCode\chockbo:

选择你要培训的课程

J2se课程
 J2ee课程

选择培训的内容:J2se J2ee

20.2.10.3 elements[]; Element 表单元素对象

<表单对象>.elements 是一个数组，包含了该表单所有的对象。一般我们不用该数组，而直接引用各个具体的对象。

20.2.10.4 Hidden 隐藏对象

由 “<input type="hidden">” 指定。

属性

name

返回/设定用<input name="...">指定的元素名称。

value

返回/设定用<input value="...">指定的元素的值。

form

返回包含本元素的表单对象。

20.2.10.5 Password 密码输入区对象

由 “<input type="password">” 指定。

属性

name

返回/设定用<input name="...">指定的元素名称。

value

返回/设定密码输入区当前的值。

defaultValue

返回用<input value="...">指定的默认值。

form

返回包含本元素的表单对象。

方法

blur()

从对象中移走焦点。

focus()

让对象获得焦点。

select()

选中密码输入区里全部文本。

事件

onchange

20.2.10.6 Radio 单选域对象

由 “<input type="radio">” 指定。一组 Radio 对象有共同的名称 (name 属性)，这样

的话，`document.formName.radioName` 就成了一个数组。要访问单个 `Radio` 对象就要用：`document.formName.radioName[x]`。

单个 `Radio` 对象的属性

name

返回/设定用 `<input name="...">` 指定的元素名称。

value

返回/设定用 `<input value="...">` 指定的元素的值。

form

返回包含本元素的表单对象。

checked

返回/设定该单选域对象是否被选中。这是一个布尔值。

defaultChecked

返回/设定该对象默认是否被选中。这是一个布尔值。

方法

blur()

从对象中移走焦点。

focus()

让对象获得焦点。

click()

模拟鼠标点击该对象。

事件

onclick

20.2.10.7 Reset 重置按钮对象

由 `<input type="reset">` 指定。因为 `Reset` 也是按钮，所以也有 `Button` 对象的属性和方法。至于 `"onclick"` 事件，一般用 `Form` 对象的 `onreset` 代替。

20.2.10.8 Select 选择区（下拉菜单、列表）对象

由 `<select>` 指定。

属性

name

返回/设定用 `<input name="...">` 指定的元素名称。

length

返回 `Select` 对象下选项的数目。

selectedIndex

返回被选中的选项的下标。这个下标就是在 `options[]` 数组中该选项的位置。如果 `Select` 对象允许多项选择，则返回第一个被选中的选项的下标。

form

返回包含本元素的表单对象。

方法

blur()

从对象中移走焦点。

focus()

让对象获得焦点。

事件

onchange

20.2.10.9 options[]; Option 选择项对象

options[] 是一个数组，包含了在同一个 Select 对象下的 Option 对象。Option 对象由 "<select>" 下的 "<options>" 指定。

options[] 数组的属性

length; selectedIndex 与所属 Select 对象的同名属性相同。

单个 *Option* 对象的属性

text

返回/指定 Option 对象所显示的文本

value

返回/指定 Option 对象的值，与<options value="...">一致。

index

返回该 Option 对象的下标。对此并没有什么好说，因为要指定特定的一个 Option 对象，都要先知道该对象的下标。

selected

返回/指定该对象是否被选中。通过指定 true 或者 false，可以动态的改变选中项。

defaultSelected

返回该对象默认是否被选中。true / false。

下拉框的取值

代码:

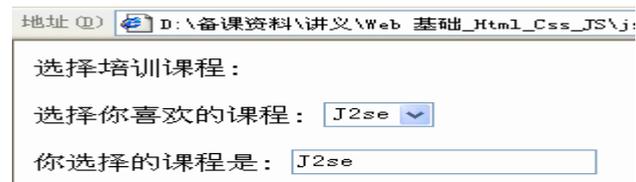
```
<html>
  <head>
    <script type="text/javascript">
      function put()
      {
        txt=document.forms[0].myClass.options[document.forms[0].myClass.selectedIndex].text
        document.forms[0].classSec.value=txt
      }
    </script>
  </head>
  <body>
    选择培训课程:<br>
    <form>
    选择你喜欢的课程:
    <select name="myClass" onchange="put()">
      <option>J2me</option>
      <option>J2se</option>
```

```

        <option>J2ee</option>
    </select>
    <br><br>
    你选择的课程是: <input type="text" name="classSec" size="20">
</form>
</body>
</html>

```

我想看一下:[jsCode\select.HTML](#)



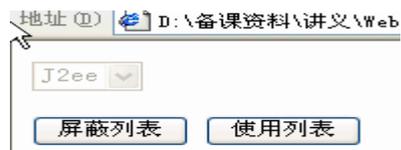
设置下拉列表的可用性

```

<html>
  <head>
    <script type="text/javascript">
      function makeDisable()
      {
        var x=document.getElementById("mySelect")
        x.disabled=true
      }
      function makeEnable()
      {
        var x=document.getElementById("mySelect")
        x.disabled=false
      }
    </script>
  </head>
  <body>
    <form>
      <select id="mySelect">
        <option>J2ee</option>
        <option>J2se</option>
        <option>J2me</option>
      </select>
      <br>
      <br>
      <input type="button" onclick="makeDisable()" value="屏蔽列表">
      <input type="button" onclick="makeEnable()" value="使用列表">
    </form>
  </body>
</html>

```

我想看一下:[jsCode\selectDisable.HTML](#)



20.2.10.10 Submit 提交按钮对象

由 “<input type="submit">”指定。因为 Submit 也是按钮，所以也有 Button 对象的属性和方法。至于 “onclick” 事件，一般用 Form 对象的 onsubmit 代替。

20.2.10.11 Text 文本框对象

由 “<input type="text">” 指定。Password 对象也是 Text 对象的一种，所以 Password 对象所有的属性、方法和事件，Text 对象都有。

20.2.10.12 Textarea 多行文本输入区对象

由 “<textarea>” 指定。Textarea 对象所有的属性、方法和事件和 Text 对象相同，也就是跟 Password 对象一样。

20.2.10.13 images[]; Image 图片对象

document.images[] 是一个数组，包含了文档中所有的图片 ()。要引用单个图片，可以用 document.images[x]。如果某图片包含 “name” 属性，也就是用 “” 这种格式定义了一幅图片，就可以使用 “document.images['...']” 这种方法来引用图片。在 IE 中，如果某图片包含 ID 属性，也就是用 “” 这种格式定义了一幅图片，就可以直接使用 “<imageID>” 来引用图片。

单个 Image 对象的属性

name; src; width; height; vspace; hspace; border 这些属性跟标记里的同名属性是一样的。这些属性最有用的就是 src 了，通过对 src 属性赋值，可以实时的更改图片。

事件

onclick

20.3 综合例子

表单验证

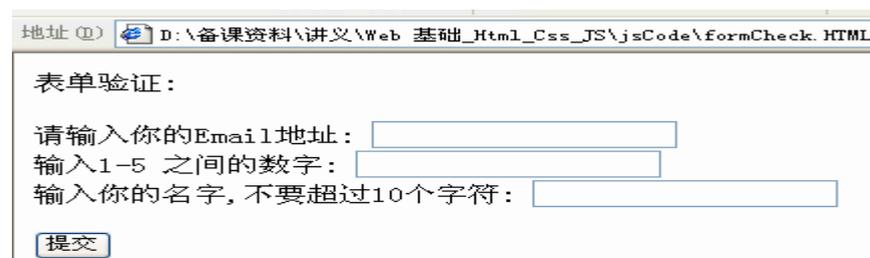
代码:

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

```
function validate()
{
    x=document.myForm
    at=x.email.value.indexOf("@")
    code=x.code.value
    firstname=x.fname.value
    submitOK="True"
    if (at==-1)
    {
        alert("不是一个有效的 EMail 地址!");
        if(x.email==null)
        {
            x.email.focus();
        }
        else
        {
            x.email.select();
        }
        return false;
    }
    if(code == null)
    {
        alert("请输入一个数字!");
        x.code.focus();
        return false;
    }
    if (code<1 || code>5)
    {
        alert("请输入 1-5 之间的数字!");
        x.code.select();
        return false;
    }
    if (firstname.length>10 || firstname.length==0)
    {
        if(firstname.length==0)
        {
            alert("你的名字必须填入!");
            x.fname.focus();
            return false;
        }
        alert("你的名字不要超过 10 个字符!")
        x.fname.select();
        return false;
    }
}
```

```
        return true;
    }
</script>
</head>
<body>
    表单验证:<br>
    <form name="myForm" action="mailto:lxz003@126.com" onsubmit="return
validate()" method="post">
        请输入你的 Email 地址: <input type="text" name="email" size="20"><br>
        输入 1-5 之间的数字: <input type="text" name="code" size="20"><br>
        输入你的名字, 不要超过 10 个字符: <input type="text" name="fname"
size="20"><br><br>
        <input type="submit" value="提交">
    </form>
</body>
</html>
```

我想看一下:[jsCode\formCheck.HTML](#)



第二十一讲 使用框架和 Cookies

我们在这一讲学习使用使用框架和 Cookies,主要内容是:

在 **JavaScript** 中使用框架
设定 **cookies**,删除 **cookies**

21.1 使用框架

每一个 HTML 文件占用一个 window 对象,包括定义框架的网页(“框架网页”)。每一个框架都是包含它的页的 window 对象的一个子对象,要引用它.可以用以下几种方法之一:

```
window.frames[x]
```

```
window.frames['frameName']
```

```
window.frameName
```

其中, x 指的是该 window 对象中指定的第几个框架,与其它数组一样, x 也是从零开始的。frameName 指的是该框架的名字,跟<frame>里的“name”属性一样。

如果使用 window.frameName 指定的 window 对象又是一个框架网页,那么引用它的框架的方法: window.frameName.subFrameName。以此类推。

要注意的是,无论在何处,引用“window”对象所返回的,都是“当前”window 对象。如果要访问其它 window 对象,就要用到 parent 和 top 属性。parent 指的是“父级”window 对象,也就是包含当前 window 对象的框架网页; top 指的是窗口最顶端的 window 对象。

21.2 使用 Cookies

在 document 对象中有一个 cookie 属性。

Cookie 的基本知识

什么是 Cookie:

某些 Web 站点在你的硬盘上用很小的文本文件存储了一些信息，这些文件就称为 Cookie。

JavaScript 提供了对 Cookies 的很全面的访问权利。

每个 Cookie 都是这样的：`<cookie 名>=<值>`

`<cookie 名>`的限制与 JavaScript 的命名限制大同小异，不能用 JavaScript 关键字，只能用可以用在 URL 编码中的字符。只要你只用字母和数字命名，就完全没有问题了。

`<值>`的要求也是“只能用可以用在 URL 编码中的字符”。

每个 Cookie 都有失效日期，一旦电脑的时钟过了失效日期，这个 Cookie 就会被删掉。我们不能直接删掉一个 Cookie，但是可以用设定失效日期早于现在时刻的方法来间接删掉它。

每个网页，或者说每个站点，都有它自己的 Cookies，这些 Cookies 只能由这个站点下的网页来访问，来自其他站点或同一站点下未经授权的区域网页，是不能访问的。每一“组”Cookies 有规定的总大小（大约 2KB 每“组”），一超过最大总大小，则最早失效的 Cookie 先被删除，来让新的 Cookie “安家”。

使用 `document.cookie` 属性

如果直接使用 `document.cookie` 属性，或者说，用某种方法，例如给变量赋值，来获得 `document.cookie` 的值，我们就可以知道在现在的文档中有多少个 Cookies，每个 Cookies 的名字，和它的值。例如，在某文档中添加“`document.write(document.cookie)`”，结果显示：`name=kevin; email=kevin@kevin.com; lastvisited=index.html`

这意味着，文档包含 3 个 Cookies：`name`、`email` 和 `lastvisited`，它们的值分别是 `kevin`、`kevin@kevin.com` 和 `index.html`。可以看到，两个 Cookies 之间是用分号和空格隔开的。

我们先用 `var cookieString = document.cookie` 得到一个存储 cookie 的字符串，我们再用 `cookieString.split(';')` 方法得到每个 Cookie 分开的一个数组。

设定一个 Cookie 的方法是对 `document.cookie` 赋值。与其它情况下的赋值不同，向 `document.cookie` 赋值不会删除掉原有的 Cookies，而只会增添 Cookies 或更改原有 Cookie。赋值的格式：

```
document.cookie = 'cookieName=' + escape('cookieValue')
+ ';expires=' + expirationDateObj.toGMTString();
```

`cookieName` 表示 Cookie 的名称，`cookieValue` 表示 Cookie 的值，`expirationDateObj` 表示储存着失效日期的日期对象名，如果不需要指定失效日期，则不需要第二行。不指定失效日期，则浏览器默认是在关闭浏览器（也就是关闭所有窗口）之后过期。

`escape()` 方法：Cookie 的值的的要求是“只能用可以用在 URL 编码中的字符”。我们知道“`escape()`”方法是把字符串按 URL 编码方法来编码的，那我们只需要用一个“`escape()`”方法来处理输出到 Cookie 的值，用“`unescape()`”来处理从 Cookie 接收过来的值就万无一失了。而且这两个方法的最常用用途就是处理 Cookies。其实设定一个 Cookie 只是“`document.cookie = 'cookieName=cookieValue'`”这么简单，但是为了避免在 `cookieValue` 中出现 URL 里不准出现的字符，还是用一个 `escape()` 好。

`expires` 设定 cookie 的失效时间，`toGMTString()` 方法：设定 Cookie 的时效日期都是用 GMT 格式的时间的，其它格式的时间是没有作用的。

21.3 使用 cookie 的例子

我们设定一个“`name=rose`”的 Cookie，在 3 个月后过期

代码:

```
var expires = new Date();
expires.setTime(expires.getTime() + 3 * 30 * 24 * 60 * 60 * 1000);
/* 三个月 x 一个月当作 30 天 x 一天 24 小时
   x 一小时 60 分 x 一分 60 秒 x 一秒 1000 毫秒 */
document.cookie = 'name=rose;expires=' + expires.toGMTString();
```

注意:

我们在这里没有使用 `escape()` 方法,这是因为我们知道 `rose` 是一个合法的 URL 编码字符串,也就是说, `'rose' == escape('rose')`。一般来说,如果设定 `Cookie` 时不用 `escape()`,那获取 `Cookie` 时也不用 `unescape()`。

编写一个函数,作用是查找指定 `Cookie` 的值

代码:

```
function getCookie(cookieName) {
    var cookieString = document.cookie;
    var start = cookieString.indexOf(cookieName + '=');
    // 加上等号的原因是避免在某些 Cookie 的值里有
    // 与 cookieName 一样的字符串。
    if (start == -1) // 找不到
        return null;
    start += cookieName.length + 1;
    var end = cookieString.indexOf(';', start);
    if (end == -1) return unescape(cookieString.substring(start));
    return unescape(cookieString.substring(start, end));
}
```

该函数在找到 `Cookie` 时,就会返回 `Cookie` 的值,否则返回 "null"。

删除刚才设定的 `name=rose` 的 `Cookie`

代码:

```
var expires = new Date();
expires.setTime(expires.getTime() - 1);
document.cookie = 'name=rose;expires=' + expires.toGMTString();
```

可以看到,只需要把失效日期改成比现在日期早一点(这里是早 1 毫秒),就可以删掉 `Cookie` 了。

第二十二讲 用 JavaScript 建立日历控件

我们在这一讲将利用 JavaScript 建立一个日历控件:

22.1 引用日历控件的 html

代码:

calendar.html

```
<html>
  <HEAD>
    <TITLE>日历控件的使用</TITLE>
    <script type="text/javascript" src="calendar.js"></script>
    <script language="javascript">
      var CalendarWebControl = new atCalendarControl();
    </script>
  </HEAD>
  <BODY>
    <form name="form1" method="post" action="">
      请输入日期:
      <input name="dateInputsec" type="text" id="dateInputsec" size="12" maxlength="12"
readonly />
      
      <br>
      日期:
      <input name="dateInput" type="text" id="dateInput" size="12" maxlength="12"
```

```

value="2003-12-22"
        onClick="CalendarWebControl.show(this,this.value);" />
        <select name="select">
            <option selected>我被 calendar 遮挡了</option>
        </select>
    </form>
</BODY>
</HTML>

```

22.2 日历控件的 js 代码

代码:

Calendar.js

```

function
    atCalendarControl(){ var
        calendar=this;
        this.calendarPad=null;
        this.prevMonth=null;
        this.nextMonth=null;
        this.prevYear=null;
        this.nextYear=null;
        this.goToday=null;
        this.calendarClose=null;
        this.calendarAbout=null;
        this.head=null;
        this.body=null;
        this.today=[];
        this.currentDate=[];
        this.sltDate;
        this.target;
        this.source;

        /***** 加入日历底板及阴影 *****/
        this.addCalendarPad=function(){
            document.write("<div                                id='divCalendarpad'
style='position:absolute;top:100;left:0;width:255;height:167;display:none;'>");
            document.write("<iframe frameborder=0 height=168 width=255></iframe>");
            document.write("<div
style='position:absolute;top:4;left:4;width:248;height:164;background-color:#336699;'></div>");
            document.write("</div>");
            calendar.calendarPad=document.all.divCalendarpad;
        }
        /***** 加入日历面板 *****/
        this.addCalendarBoard=function(){
            var BOARD=this;

```

```

var divBoard=document.createElement("div");
calendar.calendarPad.insertAdjacentElement("beforeEnd",divBoard);
divBoard.style.cssText="position:absolute;top:0;left:0;width:250px;height:166px;border:1
outset;background-color:buttonface;";

var tbBoard=document.createElement("table");
divBoard.insertAdjacentElement("beforeEnd",tbBoard);
tbBoard.style.cssText="position:absolute;top:0;left:0;width:100%;height:10px;font-size:9pt;";
tbBoard.cellPadding=0;
tbBoard.cellSpacing=1;
tbBoard.bgColor="#333333";

/***** 设置各功能按钮的功能 *****/
/***** Calendar About Button *****/ trRow =
tbBoard.insertRow(0);
calendar.calendarAbout=calendar.insertTbCell(trRow,0,"-", "center");
calendar.calendarAbout.onclick=function(){calendar.about();}
/***** Calendar Head *****/
tbCell=trRow.insertCell(1);
tbCell.colSpan=5;
tbCell.bgColor="#99CCFF";
tbCell.align="center";
tbCell.style.cssText = "cursor:default";
calendar.head=tbCell;
/***** Calendar Close Button *****/
tbCell=trRow.insertCell(2);
calendar.calendarClose = calendar.insertTbCell(trRow,2,"x", "center");
calendar.calendarClose.title="关闭";
calendar.calendarClose.onclick=function(){calendar.hide();}

/***** Calendar PrevYear Button *****/
trRow = tbBoard.insertRow(1);
calendar.prevYear = calendar.insertTbCell(trRow,0,"&lt;&lt;", "center");
calendar.prevYear.title="上一年";
calendar.prevYear.onmousedown=function(){
calendar.currentDate[0]--;

calendar.show(calendar.target,calendar.currentDate[0]+"-"+calendar.currentDate[1]+"-"+calendar.
currentDate[2],calendar.source);
}
/***** Calendar PrevMonth Button *****/
calendar.prevMonth = calendar.insertTbCell(trRow,1,"&lt;", "center");
calendar.prevMonth.title="上一月";
calendar.prevMonth.onmousedown=function(){

```

```

calendar.currentDate[1]--;
if(calendar.currentDate[1]==0){
calendar.currentDate[1]=12;
calendar.currentDate[0]--;
}

calendar.show(calendar.target,calendar.currentDate[0]+"-"+calendar.currentDate[1]+"-"+calendar.
currentDate[2],calendar.source);
}
/***** Calendar Today Button *****/
calendar.goToday = calendar.insertTbCell(trRow,2,"今天","center",3);
calendar.goToday.title="选择今天";
calendar.goToday.onclick=function(){

calendar.sltDate=calendar.currentDate[0]+"-"+calendar.currentDate[1]+"-"+calendar.currentDate[
2];
calendar.target.value=calendar.sltDate;
calendar.hide();

//calendar.show(calendar.target,calendar.today[0]+"-"+calendar.today[1]+"-"+calendar.today[2],cal
endar.source);
}
/***** Calendar NextMonth Button *****/
calendar.nextMonth = calendar.insertTbCell(trRow,3,">","center");
calendar.nextMonth.title="下一";
calendar.nextMonth.onmousedown=function(){ calendar.currentDate[1]
++;
if(calendar.currentDate[1]==13){
calendar.currentDate[1]=1;
calendar.currentDate[0]++;
}

calendar.show(calendar.target,calendar.currentDate[0]+"-"+calendar.currentDate[1]+"-"+calendar.
currentDate[2],calendar.source);
}
/***** Calendar NextYear Button *****/
calendar.nextYear = calendar.insertTbCell(trRow,4,">>","center");
calendar.nextYear.title="下一年";
calendar.nextYear.onmousedown=function(){
calendar.currentDate[0]++;

calendar.show(calendar.target,calendar.currentDate[0]+"-"+calendar.currentDate[1]+"-"+calendar.
currentDate[2],calendar.source);

```

```

}

trRow = tbBoard.insertRow(2);
var cnDateName = new Array("周日","周一","周二","周三","周四","周五","周六");
for (var i = 0; i < 7; i++)
    { tbCell=trRow.insertCell(i)
      tbCell.innerText=cnDateName[i];
      tbCell.align="center";
      tbCell.width=35;
      tbCell.style.cssText="cursor:default;border:1 solid #99CCCC;background-color:#99CCCC;";
    }

/***** Calendar Body *****/
trRow = tbBoard.insertRow(3);
tbCell=trRow.insertCell(0);
tbCell.colSpan=7;
tbCell.height=97;
tbCell.vAlign="top";
tbCell.bgColor="#F0F0F0";
var tbBody=document.createElement("table");
tbCell.insertAdjacentElement("beforeEnd",tbBody);
tbBody.style.cssText="position:relative;top:0;left:0;width:245;height:103;font-size:9pt;"
tbBody.cellPadding=0;
tbBody.cellSpacing=1;
calendar.body=tbBody;
}

/***** 加入功能按钮公共样式 *****/
this.insertTbCell=function(trRow,cellIndex,TXT,trAlign,tbColSpan){
    var tbCell=trRow.insertCell(cellIndex);
    if(tbColSpan!=undefined) tbCell.colSpan=tbColSpan;

    var btnCell=document.createElement("button");
    tbCell.insertAdjacentElement("beforeEnd",btnCell);
    btnCell.value=TXT;
    btnCell.style.cssText="width:100%;border:1 outset;background-color:buttonface;";
    btnCell.onmouseover=function(){
        btnCell.style.cssText="width:100%;border:1 outset;background-color:#F0F0F0;";
    }

    btnCell.onmouseout=function(){
        btnCell.style.cssText="width:100%;border:1 outset;background-color:buttonface;";
    }
    // btnCell.onmousedown=function(){

```

```

// btnCell.style.cssText="width:100%;border:1 inset;background-color:#F0F0F0;";
// }
btnCell.onmouseup=function(){
    btnCell.style.cssText="width:100%;border:1 outset;background-color:#F0F0F0;";
}
btnCell.onclick=function(){
    btnCell.blur();
}
return btnCell;
}
this.setDefaultDate=function(){ var
    dftDate=new Date();
    calendar.today[0]=dftDate.getYear();
    calendar.today[1]=dftDate.getMonth()+1;
    calendar.today[2]=dftDate.getDate();
}

/***** Show Calendar *****/
this.show=function(targetObject,defaultDate,sourceObject){ if(targetObject==undefined) {
    alert("未设置目标对象. \n 方法: ATCALENDAR.show(obj 目标对象,string 默认日期,obj 点击对象);\n\n 目标对象:接受日期返回值的对象.\n 默认日期:格式为\"yyyy-mm-dd\",缺省为当日日期.\n 点击对象:点击这个对象弹出 calendar,默认为目标对象.\n");
    return false;
}
else    calendar.target=targetObject;
if(sourceObject==undefined) calendar.source=calendar.target;
else calendar.source=sourceObject;

var firstDay;
var Cells=new Array();

if(defaultDate==undefined || defaultDate==""){
    var theDate=new Array();
    calendar.head.innerHTML = calendar.today[0]+"-"+calendar.today[1]+"-"+calendar.today[2];
    theDate[0]=calendar.today[0]; theDate[1]=calendar.today[1]; theDate[2]=calendar.today[2];
}
else {
    var reg=/^\d{4}-\d{1,2}-\d{2}$/
    if(!defaultDate.match(reg)){
        alert("默认日期的格式不正确\n\n 默认日期可接受格式为:'yyyy-mm-dd'");
        return;
    }
    var theDate=defaultDate.split("-");

```

```

    calendar.head.innerText = defaultDate;
}
calendar.currentDate[0]=theDate[0];
calendar.currentDate[1]=theDate[1];
calendar.currentDate[2]=theDate[2];
theFirstDay=calendar.getFirstDay(theDate[0],theDate[1]);
theMonthLen=theFirstDay+calendar.getMonthLen(theDate[0],theDate[1]);
//calendar.setEventKey();

calendar.calendarPad.style.display="";
var theRows = Math.ceil((theMonthLen)/7);
//清除旧的日历;
while (calendar.body.rows.length > 0) {
    calendar.body.deleteRow(0)
}
//建立新的日历; var n=0;day=0;
for(i=0;i<theRows;i++){ theRow=calendar.body.insertRow(i);
for(j=0;j<7;j++){
    n++;
    if(n>theFirstDay &&
    n<=theMonthLen){ day=n-theFirstDay;
    calendar.insertBodyCell(theRow,j,day);
    }

    else {
        var theCell=theRow.insertCell(j);
        theCell.style.cssText="background-color:#F0F0F0;cursor:default;";
    }
}
}

//*****调整日历位置*****//
var offsetPos=calendar.getAbsolutePos(calendar.source);//计算对象的位置;

if((document.body.offsetHeight-(offsetPos.y+calendar.source.offsetHeight-document.body.scrollTop))<calendar.calendarPad.style.pixelHeight){
    var calTop=offsetPos.y-calendar.calendarPad.style.pixelHeight;
}
else {
    var calTop=offsetPos.y+calendar.source.offsetHeight;
}

```

```

if((document.body.offsetWidth-(offsetPos.x+calendar.source.offsetWidth-document.body.scrollLeft))>calendar.calendarPad.style.pixelWidth){
    var calLeft=offsetPos.x;
}
else{
    var calLeft=calendar.source.offsetLeft+calendar.source.offsetWidth;
}
//alert(offsetPos.x);
calendar.calendarPad.style.pixelLeft=calLeft;
calendar.calendarPad.style.pixelTop=calTop;
}
/***** 计算对象的位置 *****/
this.getAbsolutePos = function(el) {
    var r = { x: el.offsetLeft, y: el.offsetTop };
    if (el.offsetParent) {
        var tmp = calendar.getAbsolutePos(el.offsetParent);
        r.x += tmp.x;
        r.y += tmp.y;
    }
    return r;
};

/***** 插入日期单元格 *****/
this.insertBodyCell=function(theRow,j,day,targetObject){
var theCell=theRow.insertCell(j);
if(j==0) var theBgColor="#FF9999"; else var
theBgColor="#FFFFFF"; if(day==calendar.currentDate[2]) var
theBgColor="#CCCCCC"; if(day==calendar.today[2]) var
theBgColor="#99FFCC"; theCell.bgColor=theBgColor;
theCell.innerText=day;
theCell.align="center";
theCell.width=35;
theCell.style.cssText="border:1 solid #CCCCCC;cursor:hand;";
theCell.onmouseover=function(){ theCell.bgColor="#
FFFFCC"; theCell.style.cssText="border:1
outset;cursor:hand;";
}
theCell.onmouseout=function(){
theCell.bgColor=theBgColor;
theCell.style.cssText="border:1 solid #CCCCCC;cursor:hand;";
}
theCell.onmousedown=function(){
theCell.bgColor="#FFFFCC";

```

```

    theCell.style.cssText="border:1 inset;cursor:hand;";
}
theCell.onclick=function(){
    if(calendar.currentDate[1].length<2) calendar.currentDate[1]="0"+calendar.currentDate[1];
    if(day.toString().length<2) day="0"+day;
    calendar.sltDate=calendar.currentDate[0]+"-"+calendar.currentDate[1]+"-"+day;
    calendar.target.value=calendar.sltDate;
    calendar.hide();
}
}
/***** 取得月份的第一天为星期几 *****/
this.getFirstDay=function(theYear, theMonth){
    var firstDate = new Date(theYear,theMonth-1,1);
    return firstDate.getDay();
}
/***** 取得月份共有几天 *****/

this.getMonthLen=function(theYear, theMonth) {
    theMonth--;
    var oneDay = 1000 * 60 * 60 * 24;
    var thisMonth = new Date(theYear, theMonth, 1);
    var nextMonth = new Date(theYear, theMonth + 1, 1);
    var len = Math.ceil((nextMonth.getTime() - thisMonth.getTime())/oneDay);
    return len;
}
/***** 隐藏日历 *****/
this.hide=function(){
    //calendar.clearEventKey();
    calendar.calendarPad.style.display="none";
}
/***** 从这里开始 *****/
this.setup=function(defaultDate){ calendar.addCalendarPad()
;
    calendar.addCalendarBoard();
    calendar.setDefaultDate();
}
/***** 关于 AgetimeCalendar *****/
this.about=function(){
    /**/alert("Agetime Calendar V1.0\n\nwww.agetime.com\n");
    popLeft = calendar.calendarPad.style.pixelLeft+4;
    popTop = calendar.calendarPad.style.pixelTop+25;
    var popup = window.createPopup();
    var popupBody = popup.document.body;
    popupBody.style.cssText="border:solid 2 outset;font-size:9pt;background-color:#F0F0F0;";
}

```

```
var popHtml = "<span style='color:#336699;font-size:12pt;'><U> 关 于  
AgetimeCalendar</U></span><BR><BR>"; popHtml+="版  
本: v1.0<BR>日期: 2004-03-13";  
popupBody.innerHTML=popHtml;  
popup.show(popLeft,popTop,240,136,document.body); */  
var strAbout = "About AgetimeCalendar\n\n";  
strAbout+="-\t: 关于\n";  
strAbout+="x\t: 隐藏\n";  
strAbout+="<<\t: 上一年\n";  
strAbout+="<\t: 上一月\n";  
  
strAbout+="今日\t: 返回当天日期\n";  
strAbout+=">\t: 下一月\n";  
strAbout+="<<\t: 下一年\n";  
strAbout+="\n\nAgetimeCalendar\nVersion:v1.0\nDesigned By: 暂停打印 2004-03-16  
afos_koo@hotmail.com \n";  
alert(strAbout);  
}  
  
calendar.setup();  
}
```

第四部分 综合应用:建立你自己的网站

第二十三 讲建立你自己的网站

23.1 准备工作

我们使用的网页制做工具是 dreamwave MX2004,我们还要准备好建站的素材,包括图片等,构思自己的网站,我们的网站暂时命名为”听雨轩”,主要的栏目包括相识雨轩,轻松一刻,雁过留声,美文欣赏,友情链接几个部分.

在网页头部,和左侧使用连接构成菜单,在顶部使用版权声明信息,右侧为友情连接信息,顶部是网站或个人主页的 logo 信息.

相识雨轩栏目,介绍听雨轩的由来,轻松一刻栏目是一个笑话的连接,雁过留声栏目,是用户留言的表单,以邮件的方式发送给网站的作者,美文欣赏主要是优美的散文.我们为各个栏目收集相关的素材和资料.为建立网页作准备.

23.2 网站目录结构

网站放在 rainRoom 文件夹下面根目录是 Html 文件,在 image 放置图片素材,css 放置样式表信息.



23.3 制做主页

主页部分使用框架把屏幕划分为五个部分,上部为网站 logo 部分,以及网站的菜单项,下面部分为版权信息说明,左侧为栏目连接,右侧为友情连接,中间为正文显示区。

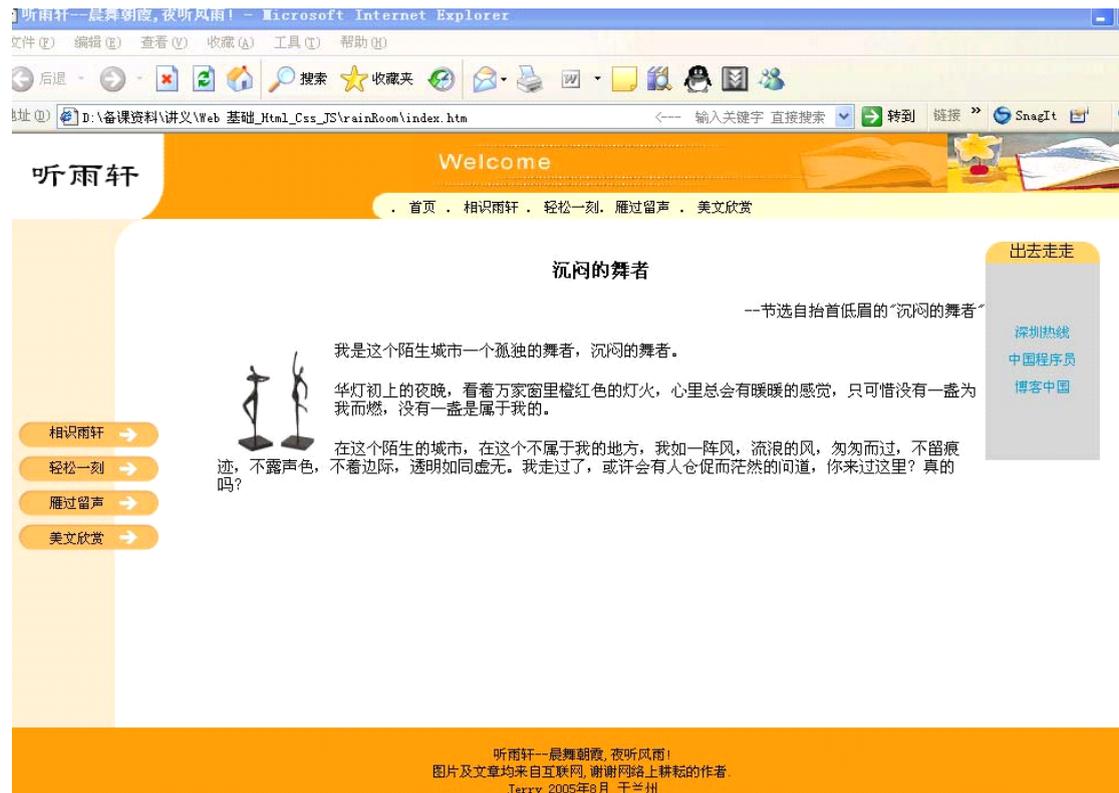
代码:

Index.htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>听雨轩--晨舞朝霞,夜听风雨!</title>
</head>

<frameset rows="75,*,75" cols="*" framespacing="0" frameborder="NO" border="0">
  <frame src="top.htm" name="topFrame" scrolling="NO" noresize >
  <frameset rows="*" cols="191*,670*,90*" framespacing="0" frameborder="NO" border="0">
    <frame src="menu.htm">
    <frame src="content.htm" name="mainFrame">
    <frame src="friendLink.htm">
  </frameset>
  <frame src="bottom.htm" name="bottomFrame" scrolling="NO" noresize>
</frameset>
<noframes><body>
</body></noframes>
</html>
```

网页效果:



23.4 制做相识雨轩栏目

代码:

About.htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>关于听雨轩!</title>
<style type="text/css">
<!--
body {
margin-left: 10px;
margin-top: 10px;
margin-right: 10px;
margin-bottom: 10px;
}
-->
</style>
<link href="css/myStyle.css" rel="stylesheet" type="text/css">
</head>
```

```

<body>
<p align="center"><strong>关于听雨轩</strong></p>
<blockquote>
  <blockquote>
    <blockquote>
      <p align="left">
      <span class="bodyText">感谢大家光临听雨轩, 能够在这里相逢,点一抹烛光,品一盏香茗,
      倾听夜雨袭来.</span></p>
      <p align="left"><span class="bodyText">在风雨中寻求恬适,在烛光中感受宁静,在香茗
      的苦涩和回甘里感悟人生,这就是听雨轩的由来.</span></p>
      <p align="left"><span class="bodyText">也希望能在这里修整疲惫的精神,在每一个朝
      霞满天的清晨舞动你的希望和理想.</span></p>
    </blockquote>
  </blockquote>
  <blockquote>
    <p class="bodyText"> 晨舞朝霞,夜听风雨!</p>
  </blockquote>
</blockquote>
</blockquote>
</body>
</html>

```

网页效果:



关于听雨轩



感谢大家光临听雨轩, 能够在这里相逢,点一抹烛光,品一盏香茗,倾听夜雨袭来.

在风雨中寻求恬适,在烛光中感受宁静,在香茗的苦涩和回甘里感悟人生,这就是听雨轩的由来.

也希望能在这里修整疲惫的精神,在每一个朝霞满天的清晨舞动你的希望和理想.

晨舞朝霞,夜听风雨!

23.5 制做轻松一刻栏目

代码:

Joke.htm

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>无标题文档</title>
<style type="text/css">
<!--
body {
    margin-left: 10px;
    margin-top: 10px;
    margin-right: 10px;
    margin-bottom: 10px;
}
-->
</style>
<link href="css/myStyle.css" rel="stylesheet" type="text/css">
</head>

<body>
<p align="center"><strong>程序员和工程师</strong></p>
<p><span class="bodyText">
一个程序员和一名工程师同乘一架飞机而且座位正好相邻,这个程序员问工程师是否想玩一个有趣的游戏。</span></p>
<p><span class="bodyText">可是工程师正疲倦不堪,只想睡觉,于是他礼貌地拒绝了 this 提议。<br>
可是程序员紧逼不放还说这个游戏非常简单:“我问一个问题,如果你回答不出来就给我 5 美元,然后你再问我一个问题,如果我回答不出来,就给你 5 美元,如何?” 可是工程师还是礼貌地拒绝了,并且准备睡觉。</span></p>
<p><span class="bodyText"><br>
程序员有点儿着急:“好吧!那这样吧,如果你不知道答案你给我 5 美元,但是如果我不知道答案,那我给你 50 美元,怎么样?” 这下工程师有了点儿兴趣,于是同意开始游戏。程序员首先发问:“从地球到月球的距离有多远?” </span></p>
<p><span class="bodyText">工程师挠挠头,给了程序员 5 美元。现在,轮到工程师了。他问程序员:“什么东西用 3 条腿上山,用 4 条腿下山?”;程序员瞠目结舌地看着他,然后拿出了他的笔记本电脑,开始查阅 he 所有的资料。</span></p>
<p><span class="bodyText">一个小时后,他叫醒了工程师,给了工程师 5 0 美元。工程师礼貌地接过 5 0 美元,然后转过头想接着睡觉。“嘿!答案到底是什么呀?” 程序员有点儿不高兴。工程师一言不发,拿出 5 美元给了程序员,然后倒头便睡。</span></p>
</body>
</html>
网页效果:

```

· 首页 · 相识雨轩 · 轻松一刻 · 雁过留声 · 美文欣赏

程序员和工程师



一个程序员和一名工程师同乘一架飞机而且座位正好相邻，这是一个有趣的游戏。

可是工程师正疲倦不堪，只想睡觉，于是他礼貌地拒绝了这个游戏。可是程序员紧逼不放还说这个游戏非常简单：“我问一个问题，5美元，然后你再问我一个问题，如果我回答不出来，就给你5美元还是礼貌地拒绝了，并且准备睡觉。

程序员有点儿着急：“好吧！那这样吧，如果你不知道答案你给我5美元，但如果你知道答案你给我50美元，怎么样？”这下工程师有了点儿兴趣，于是同意开始游戏。程序员问：“月球到地球的距离有多远？”

工程师挠挠头，给了程序员5美元。现在，轮到工程师了。他问程序员：“你有多少条腿下山？”程序员瞠目结舌地看着他，然后拿出了他的笔记本电脑，开始

一个小时后，他叫醒了工程师，给了工程师50美元。工程师礼貌地接过50美元，然后倒头便睡。工程师问：“嘿！答案到底是什么呀？”程序员有点儿不高兴。工程师一言不发，然后倒头便睡。

23.6 制做雁过留声栏目

代码:

Contact.htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>雁过留声</title>
<style type="text/css">
<!--
body {
    background-image: url(image/butterfly.gif);
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
-->
</style>
<link href="css/myStyle.css" rel="stylesheet" type="text/css">
</head>
```

```
<body>
<form action="mailto:lxz003@126.com" method="post" name="contact" id="contact">
  <table width="50%" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
      <td height="80" colspan="2"><div align="center"><strong> 过 客 留 言
</strong></div></td>
    </tr>
    <tr>
      <td width="20%" height="30" class="bodyText">昵称:</td>
      <td width="80%"><input name="userName" type="text" class="inputBgcolor"
id="userName" size="50" maxlength="10"></td>
    </tr>
    <tr>
      <td height="30" class="bodyText">Email:</td>
      <td><input name="Email" type="text" class="inputBgcolor" id="Email" size="50"
maxlength="30"></td>
    </tr>
    <tr>
      <td height="80" class="bodyText">留言:</td>
      <td><textarea name="guessWord" cols="48" rows="8" class="inputBgcolor"
id="guessWord"></textarea></td>
    </tr>
    <tr>
      <td height="30">&nbsp;</td>
      <td><div align="center">
        <input name="Submit" type="submit" class="button" value="提交">
        <input name="Submit" type="reset" class="button" value="重置">
      </div></td>
    </tr>
  </table>
</form>
</body>
</html>
```

网页效果:

. 首页 . 相识雨轩 . 轻松一刻 . 雁过留声 . 美文欣赏

过客留言

昵称:

Email:

留言:

23.7 制做美文欣赏栏目

代码:

goodArt.htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>美文欣赏</title>
<style type="text/css">
<!--
body {
    margin-left: 10px;
    margin-top: 10px;
    margin-right: 10px;
    margin-bottom: 10px;
}
-->
</style>
<link href="css/myStyle.css" rel="stylesheet" type="text/css">
</head>

<body>
<blockquote>
```



```

        <td width="95%"></td>
    </tr>
    <tr bgcolor="#fffecc">
        <td colspan="2" nowrap> . <a href="index.htm" title="听雨轩首页"
target="_top" class="topMenu"> 首页 </a> . <a href="about.htm" title="想了解听雨轩吗"
target="mainFrame" class="topMenu"> 相识雨轩 </a>. <a href="joke.htm" title="心情感触"
target="mainFrame" class="topMenu"> 轻松一刻</a>. <a href="contact.htm" title="给听雨轩留
言" target="mainFrame" class="topMenu"> 雁过留声</a> . <a href="goodArt.htm" title="优美
经典的文字" target="mainFrame" class="topMenu">美文欣赏</a></td>
    </tr>
</table></td>
</tr>
</table>
</body>
</html>

```

网页效果:



23.9 左侧菜单

代码:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>无标题文档</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-bottom: 0px;
    margin-right: 0px;
}
-->
</style>
<link href="css/myStyle.css" rel="stylesheet" type="text/css">
</head>

```

```

<body>
<table width="139" height="421" border="0" align="left" cellpadding="0" cellspacing="0">
  <tr>
    <td height="26"></td>
  </tr>
  <tr>
    <td height="100%" nowrap background="image/08.gif"><table width="139" height="122"
border="0" cellpadding="0" cellspacing="0">
      <tr>
        <td width="124" height="122" background="image/07.gif"><table width="119"
height="119" border="0" cellpadding="0" cellspacing="0">
          <tr>
            <td width="44" height="31">&nbsp;</td>
            <td width="75" valign="middle" nowrap><a href="about.htm"
target="mainFrame" class="sideMenu">相识雨轩</a></td>
          </tr>
          <tr>
            <td width="44" height="31">&nbsp;</td>
            <td valign="middle" nowrap><a href="joke.htm" target="mainFrame"
class="sideMenu">轻松一刻</a></td>
          </tr>
          <tr>
            <td width="44" height="31">&nbsp;</td>
            <td valign="middle" nowrap><a href="contact.htm" target="mainFrame"
class="sideMenu">雁过留声</a></td>
          </tr>
          <tr>
            <td width="44" height="31">&nbsp;</td>
            <td valign="middle" nowrap><a href="goodArt.htm" target="mainFrame"
class="sideMenu">美文欣赏</a></td>
          </tr>
        </table></td>
      </tr>
    </table></td>
  </tr>
</table>
</body>
</html>

```

网页效果:



23.10 首页内容

代码:

Content.htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>听雨轩--晨舞朝霞,夜听风雨!</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
-->
</style>
<link href="css/myStyle.css" rel="stylesheet" type="text/css">
</head>

<body>
<p align="center">&nbsp;&nbsp;&nbsp;</p>
```

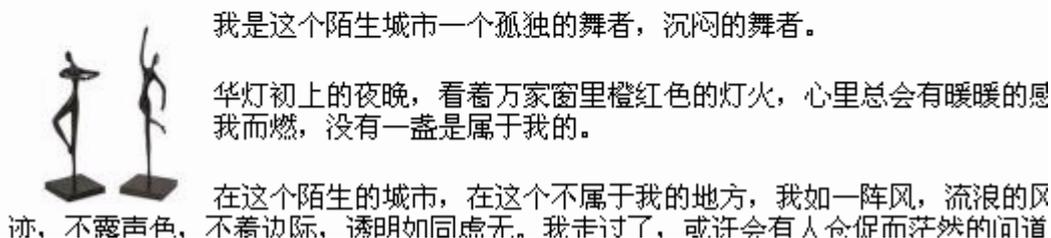
```

<p align="center"><strong>沉闷的舞者</strong></p>
<p align="right"> <span class="bodyText">-- 节选自抬首低眉的 &quot;沉闷的舞者
&quot;</span></p>
<p class="bodyText">我
是这个陌生城市一个孤独的舞者，沉闷的舞者。
</p>
<p class="bodyText"> 华灯初上的夜晚，看着万家窗里橙红色的灯火，心里总会有暖暖的感觉，
只可惜没有一盏为我而燃，没有一盏是属于我的。</p>
<p><span class="bodyText">在这个陌生的城市，在这个不属于我的地方，我如一阵风，流浪
的风，匆匆而过，不留痕迹，不露声色，不着边际，透明如同虚无。我走过了，或许会有人
仓促而茫然的问道，你来过这里？真的吗？</span><br>
</p>
</body>
</html>
网页效果:

```

沉闷的舞者

--节选自:



23.11 友情连接

代码:

friendLink:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>无标题文档</title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
margin-right: 0px;
margin-bottom: 0px;

```

```

}
-->
</style>
<link href="css/myStyle.css" rel="stylesheet" type="text/css">
</head>

<body>
<table width="63%" height="211" border="0" cellspacing="0" class="friendLink">
  <tr>
    <td width="100%" height="15">&nbsp;</td>
  </tr>
  <tr>
    <td height="17" background="image/links_up.gif"><div align="center" class="bodyText">
出去走走</div></td>
  </tr>
  <tr>
    <td height="171" bgcolor="#CCCCCC"><table width="100%" border="0">
      <tr>
        <td width="100%" height="20" nowrap><div align="center"><a
href="http://www.szonline.net" title="深圳热线" target="_blank" class="friendLink">深圳热线
</a></div></td>
      </tr>
      <tr>
        <td height="20" nowrap><div align="center"><a href="http://www.csdn.net" title="中
国程序员" target="_blank" class="friendLink">中国程序员</a></div></td>
      </tr>
      <tr>
        <td height="20" nowrap><div align="center"><a href="http://www.bokee.com" title="
博客中国" target="_blank" class="friendLink">博客中国</a></div></td>
      </tr>
    </table></td>
  </tr>
</table>
</body>
</html>

```

网页效果:



23.12 版权信息

代码:

Bottom.htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>听雨轩--晨舞朝霞,夜听风雨!</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-bottom: 0px;
    margin-right: 0px;
}
-->
</style>
<link href="css/myStyle.css" rel="stylesheet" type="text/css">
</head>

<body>
<table width="100%" border="0" cellpadding="0" cellspacing="0" bgcolor="#ff9900">
  <tr>
    <td height="15" nowrap>&nbsp;</td>
  </tr>
  <tr>
    <td height="15" nowrap><div align="center" class="bottomText">听雨轩--晨舞朝霞,夜听
风雨!</div></td>
  </tr>
  <tr>
    <td height="15" nowrap><div align="center" class="bottomText">图片及文章均来自互联
网,谢谢网络上耕耘的作者.</div></td>
  </tr>
  <tr>
    <td height="15" nowrap><div align="center" class="bottomText">Jerry 2005 年 8 月 于兰
州</div></td>
  </tr>
  <tr>
    <td height="15" nowrap>&nbsp;</td>
  </tr>
</table>
```

</body>

</html>

网页效果:

听雨轩--晨舞朝霞,夜听风雨!
图片及文章均来自互联网,谢谢网络上耕耘的作者.
Jerry 2005年8月 于兰州

23.13 制做样式表

代码:

myStyle.css

```
/* CSS Document */
```

```
A.topMenu{
```

```
    text-decoration:none;
```

```
    font-size: 12px;
```

```
    color: #000000;
```

```
}
```

```
A.sideMenu{
```

```
    text-decoration: none;
```

```
    font-size: 12px;
```

```
    color: #000000;
```

```
}
```

```
A.friendLink{
```

```
    text-decoration: none;
```

```
    font-size: 12px;
```

```
    color: #0099CC;
```

```
}
```

```
.button{
```

```
    FONT-SIZE: 12px; LINE-HEIGHT: 15px;
```

```
}
```

```
.inputBgcolor{
```

```
    BACKGROUND-COLOR: #FFCCFF}
```

```
.bottomText{
```

```
    font-size: 12px;
```

```
}
```

```
.bodyText{
```

```
    font-size: 14px;
```

```
}
```

23.14 网站的上传

我们接下来在网上查找免费的自由空间上传我们网页的内容,我们就可以拥有自己的主页了,而且我们的主页可以和大家以前分享.

23.15 课程总结

通过学习 Html 建立我们的网页文档,使用 css 美化我们的网页,使用 JavaScript 和用户进行交互,最终我们建立了一个自己的网站“听雨轩”.我们的课程就到此结束了,对于如何建立网站的内容我们可以参照教材上的源代码,自己动手建立自己的网站,分享自己的成果.

Jerry 2005-8-26 于兰州