



校企双元合作开发“互联网+教育”新形态一体化系列教材

Java Web 应用开发案例与实战

主 编 冯建平 霍婕婷 田 智
副主编 徐 多 姜 艳 海 川 张 兴
参 编 秦艳梅 冯子龙 赵东伟 李志君
金竹茹 李金山 武 涛 张玉川



合肥工业大学出版社
HEFEI UNIVERSITY OF TECHNOLOGY PRESS

图书在版编目(CIP)数据

Java Web应用开发案例与实战 / 冯建平, 霍婕婷, 田智主编. —合肥: 合肥工业大学出版社, 2023.3

ISBN 978-7-5650-6217-9

I. ①J… II. ①冯… ②霍… ③田… III. ①JAVA语言—程序设计 IV. ①TP312.8

中国国家版本馆 CIP 数据核字(2023)第 040792 号

Java Web 应用开发案例与实战

JAVA WEB YINGYONG KAIFA ANLI YU SHIZHAN

冯建平 霍婕婷 田智 主编

责任编辑 张 慧
出版发行 合肥工业大学出版社
地 址 (230009)合肥市屯溪路 193 号
网 址 www.hfutpress.com.cn
电 话 人文社科出版中心: 0551-62903205
营销与储运管理中心: 0551-62903198
规 格 787 毫米 × 1092 毫米 1/16
印 张 19
字 数 340 千字
版 次 2023 年 3 月第 1 版
印 次 2023 年 3 月第 1 次印刷
印 刷 廊坊市广阳区九洲印刷厂
书 号 ISBN 978-7-5650-6217-9
定 价 58.00 元

如果有影响阅读的印装质量问题, 请与出版社营销与储运管理中心联系调换



随着互联网的飞速发展和日益普及，网络应用越来越多。在众多提供动态内容的Web编程技术中，Java Web因拥有各种功能强大的框架而成为主流技术，如Spring、Hibernate等。

本书以Java Web为基础，紧紧围绕最新的Java Web技术精髓展开深入讲解，以清晰的思路、精炼的实例和任务使读者快速入门，并逐步掌握Java Web编程的知识。本书注重基础理论与实用开发相结合，突出应用编程思想与开发方法的介绍，所选实例和任务均具有较强的概括性和实际应用价值。

本书是作者根据多年从事Java Web开发工作经验和讲授计算机专业相关课程的教学实践，在已编多部讲义和教材的基础上编写而成的；内容充实，循序渐进，选材上注重系统性、先进性和实用性；注重实践性，精选大量例题，所有例题已在Eclipse上调试通过，可直接引用，读者也可按照书中提示步骤自己动手完成。

本书分为12章。

第1章为Java Web概述，主要介绍了Java Web的环境搭建、如何在搭建的环境中创建Java Web项目等内容。

第2章为HTML和JavaScript基础，主要介绍了HTML和JavaScript的一些基础内容。

第3章为JSP基本语法，主要介绍了如何编写第一个JSP程序、JSP基础语法以及JSP内置对象等内容。

第4章为JSP访问数据库，主要介绍了JDBC技术和SQL语言等内容。

第5章为Servlet编程，主要介绍了如何编写第一个Servlet程序、Servlet的生命周期、使用HttpServletResponse类实现响应和使用HttpServletRequest类获得请求消息等内容。

第6章为EL和JSTL，主要介绍了使用EL、如何编写第一个JSTL程序、使用JSTL的核心标签库以及其他标签库等内容。

第7章为Ajax入门，主要介绍了XMLHttpRequest对象和Ajax与XML等内容。

第8章为Struts 2框架基础，主要介绍了下载Struts 2、了解目录结构、如何编写第一个Struts 2程序、深入理解Struts 2的配置元素、Struts 2的Action、Action访问Servlet API以及处理



结果等内容。

第 9 章为 Hibernate 框架基础，主要介绍了下载和了解 Hibernate 目录结构、如何编写第一个 Hibernate 程序、配置文件介绍、Session 的基本应用以及 Hibernate 查询等内容。

第 10 章为 Spring 框架基础，主要介绍了下载和了解 Spring 目录结构、如何编写第一个 Spring 程序、配置文件介绍、控制反转 (IoC) 以及装配 Bean 等内容。

第 11 章为 Spring MVC，主要介绍了如何编写第一个 Spring MVC 程序、Spring MVC 常用注解、Spring MVC 传递参数以及 Spring MVC 表单标签等内容。

第 12 章为在线服装订购系统，主要介绍了如何创建一个在线服装订购系统的大项目。

由于编者水平有限，书中难免存在错误和疏漏之处，希望广大读者批评指正。

编 者



第1章 Java Web概述 1

知识入门

循序渐进

1.1 搭建Java Web开发环境 2

1.2 创建第一个Java Web程序 13

知识拓展

本章习题

第2章 HTML和JavaScript基础 21

知识入门

循序渐进

2.1 HTML基础 23

2.2 JavaScript基础 33

知识拓展

本章习题

第3章 JSP基本语法 51

知识入门

循序渐进

3.1 编写第一个JSP程序 54

3.2 JSP基础语法 55

3.3 JSP内置对象 62

知识拓展

本章习题

第4章 JSP访问数据库 73

知识入门

循序渐进

4.1 JDBC技术 76

4.2 SQL语言 87

知识拓展

本章习题

**第 5 章 Servlet 编程 99**

知识入门

循序渐进

5.1 编写第一个Servlet程序 101

5.2 Servlet的生命周期 104

5.3 使用HttpServletResponse类实现响应 105

5.4 使用HttpServletRequest类获得请求消息 108

知识拓展

本章习题

第 6 章 EL和JSTL 119

知识入门

循序渐进

6.1 使用EL 121

6.2 编写第一个JSTL程序 129

6.3 使用JSTL的核心标签库(Core Tag Library) 131

6.4 其他标签库 137

知识拓展

本章习题

第 7 章 Ajax入门 149

知识入门

循序渐进

7.1 XMLHttpRequest对象 151

7.2 Ajax与XML 159

知识拓展

本章习题

第 8 章 Struts 2 框架基础 169

知识入门

循序渐进

8.1 下载Struts 2 171

8.2 了解目录结构 172

8.3 编写第一个Struts 2程序 174

8.4 深入理解Struts 2的配置元素 178

8.5 Struts 2的Action 182

8.6 Action访问Servlet API 185

8.7 处理结果 190

知识拓展

本章习题

**第 9 章 Hibernate 框架基础 199**

知识入门

循序渐进

- 9.1 下载和了解Hibernate目录结构 201
- 9.2 编写第一个Hibernate程序 203
- 9.3 配置文件介绍 208
- 9.4 Session的基本应用 214
- 9.5 Hibernate查询 220

知识拓展

本章习题

第 10 章 Spring 框架基础 229

知识入门

循序渐进

- 10.1 下载和了解Spring目录结构 232
- 10.2 编写第一个Spring程序 234
- 10.3 配置文件介绍 238
- 10.4 控制反转 (IoC) 239
- 10.5 装配Bean 243

知识拓展

本章习题

第 11 章 Spring MVC 257

知识入门

循序渐进

- 11.1 编写第一个Spring MVC程序 259
- 11.2 Spring MVC常用注解 262
- 11.3 Spring MVC传递参数 265
- 11.4 Spring MVC表单标签 271

知识拓展

本章习题

第 12 章 在线服装订购系统 277

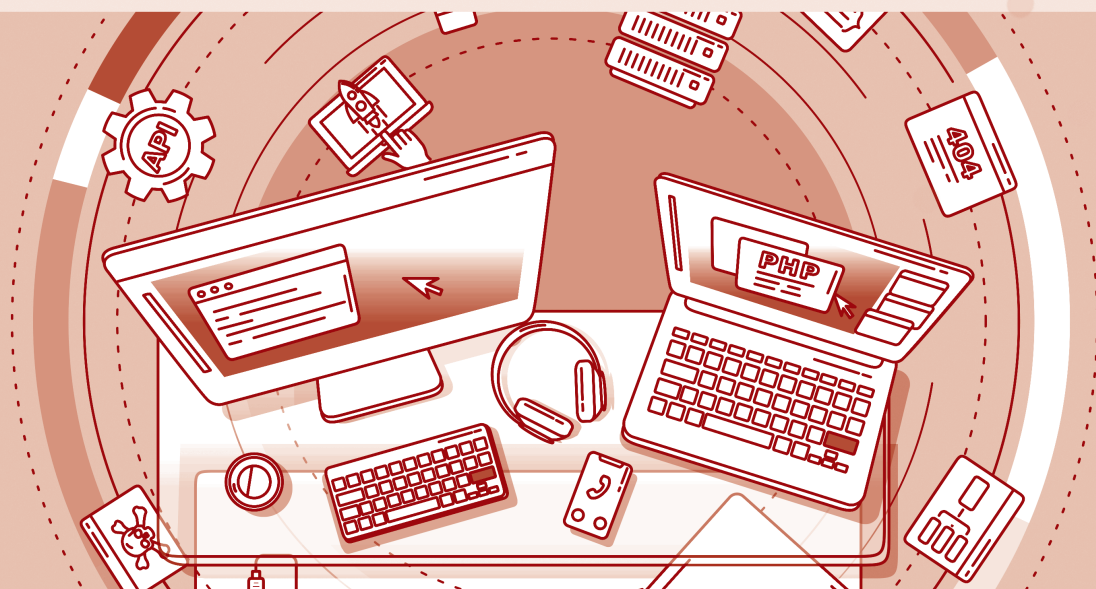
- 12.1 准备工作 278
- 12.2 程序的功能介绍 278
- 12.3 数据库设计 280
- 12.4 类文件的代码编写 281
- 12.5 编写配置文件 285
- 12.6 JSP文件的代码编写 286

参考文献 296

第2章

HTML和JavaScript基础

一个网站由很多的网页组成，而网页又是由各种元素组成的，如文本、图片、链接等。这些元素基本上是通过HTML语言和JavaScript实现的。本章将为开发者介绍关于HTML的基础内容，其中包括文档的基本结构、页面格式化标签、文字样式标记标签等。





1. HTML 介绍

HTML (hyper text markup language, 超文本标记语言), 是一种标记语言, 由 Web 的发明者蒂姆·伯纳斯·李 (Tim Berners-Lee) 和同事丹尼尔·W. 科诺里 (Daniel W. Connolly) 于 1990 年创立。它包括一系列标签, 通过这些标签可以将网络上的文档格式统一, 使分散的 Internet 资源连接为一个逻辑整体。用 HTML 编写的超文本文档称为 HTML 文档, 它能独立于各种操作系统平台。HTML 具有以下几个特点。

(1) 简易性: 超文本标记语言版本升级采用超集方式, 从而更加灵活方便。

(2) 可扩展性: 超文本标记语言的广泛应用带来了加强功能、增加标识符等要求, 超文本标记语言采取子类元素的方式, 为系统扩展带来保证。

(3) 平台无关性: 超文本标记语言可以广泛使用在各类平台上, 如 Windows、MAC 等, 这也是万维网 (WWW) 盛行的另一个原因。

(4) 通用性: HTML 是网络通用语言, 是一种简单、通用的全置标记语言。它允许网页制作者建立文本与图片相结合的页面, 这些页面可以被任何人浏览到, 无论使用的是什么类型的计算机或浏览器。

2. HTML 文档的基本结构

HTML 文档的基本结构如图 2.1 所示。

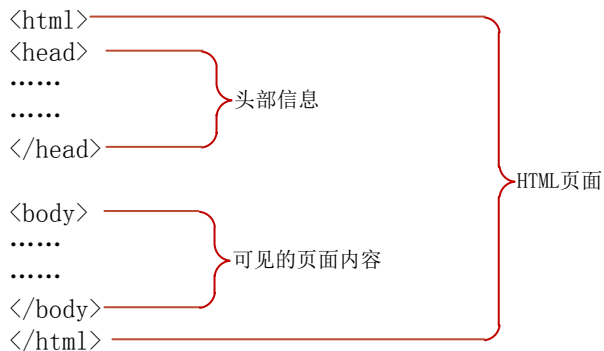


图 2.1 HTML 文档的基本结构

<head></head> 之间的内容是用来设置网页中头部信息的; <body></body> 之间的内容是浏览器中可见的页面内容。

3. HTML 的标签

带有 “<>” 符号的元素被称为 HTML 标签, 标签可以分为双标签和单标签。

(1) 单标签: 只需在尖括号中输入标记名即可, 一种单标签表示一种功能, 即没有范围之分, 其语法形式如下:



<标签名 />

(2) 双标签: 指由开始和结束两个标签符组成的标签, 其语法形式如下:

<标签名>内容</标签名>

其中, 开始标签也被称为开放标签, 结束标签也被称为闭合标签。

4. 超链接

超链接是指从一个网页指向一个目标的连接关系, 这个目标可以是另一个网页, 也可以是同一网页上的不同位置, 还可以是一个图片、一个电子邮件地址、一个文件, 甚至是一个应用程序。在一个网页中用来超链接的对象, 可以是一段文本或是一个图片。

5. JavaScript 介绍

JavaScript 在 1995 年由 Netscape 公司的 Brendan Eich 在网景导航者浏览器上首次设计实现而成。JavaScript 主要用于交互式操作、表单验证、控制文档的外观和内容、对浏览器控制、与 HTML 表单交互、用 Cookie 读写客户的状态、网页特效、能够嵌入任何应用程序和服务器脚本开发等方面。它主要具有基于对象语言、变量没有明确的类型、简单、易学、易用、跨平台等特点。



2.1

HTML 基础

本节将对 HTML 的一些基础内容进行讲解, 如 HTML 页面格式化标签、文本格式标签、超链接标签等。

2.1.1 HTML 页面格式化标签

本节将讲解与页面格式相关的标签, 包括标题标签、段落标签、水平线标签及换行标签。



HTML 页面
格式化标签

1. 标题标签

在网页中使用 <h> 标签来定义标题, 其语法如下:

<h#>标题内容 </h#>

其中, “#” 是数字 1~6 中任意的一个, 系统一共提供 6 种标题大小, 从 1 到 6, 数字越大, 标题字号就越小。



实例 2-1 下面将字号由小到大排列，具体操作步骤如下：

- (1) 打开创建的 Java Web 项目。
- (2) 创建并添加一个 HTML 文件，命名为 NewFile.html。
- (3) 打开 NewFile.html 文件，在此文件实现使用标签将字号由小到大排列，具体代码如下：

```
<body>
<h6>标题 6</h6>
<h5>标题 5</h5>
<h4>标题 4</h4>
<h3>标题 3</h3>
<h2>标题 2</h2>
<h1>标题 1</h1>
</body>
```

运行程序，效果如图 2.2 所示。

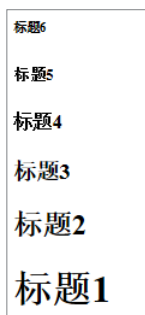


图 2.2 效果

2. 段落标签

在网页中使用 `<p>` 标签来定义段落。`<p>` 标签是 HTML 文档中最常见的标签，默认情况下，文本在一个段落中会根据浏览器窗口的大小自动换行，其语法形式如下：

```
<p>段落文字</p>
```

3. 水平线标签

在网页中常常看到一些水平线将段落与段落隔开，这些水平线可以通过 `<hr>` 标签实现，其语法形式如下：

```
<hr>
```

在网页中输入一个 `<hr>`，即可添加一条默认样式的水平线。

开发者可以使用表 2.1 中的属性对显示的水平线进行设置。

表 2.1 设置水平线

属性名	含义	属性值
align	设置水平线的对齐方式	可选择 left、right、center 三个值，默认为 center，居中对齐显示



续表

属性名	含义	属性值
size	设置水平线的粗细	以像素为单位，默认为 2 像素
color	设置水平线的颜色	可用颜色名称、十六进制 #RGB、rgb(r,g,b)
width	设置水平线的宽度	可以是确定的像素值，也可以是浏览器窗口的百分比，默认为 100%

4. 换行标签

在网页中，如果想要将某段文本强制换行显示，就需要使用换行标签
，其语法形式如下：

```
<br>
```

2.12 HTML 文本格式标签

文本格式标签可以用来控制网页中文本的字体、字号、颜色和样式。其中，样式包含粗体、斜体、下划线、删除线、上标和下标。本节将详细介绍这些标签。



HTML 文本格式标签

1. 字体、字号和颜色

在 HTML 中，对文本字体、字号和颜色的设置需要使用 标签，其语法形式如下：

```
<font 属性="属性值">文本内容</font>
```

其中，属性包含 3 个，介绍如下。

- (1) face: 设置文字的字体，如微软雅黑、黑体、宋体等。
- (2) size: 设置文字的大小，可以取 1~7 的整数值。
- (3) color: 设置文字的颜色。

实例 2-2 下面将实现对显示的文本显示不同的字体、字号和颜色，具体的操作步骤如下：

- (1) 打开创建的 Java Web 项目。
- (2) 创建并添加一个 HTML 文件，命名为 NewFile.html。
- (3) 打开 NewFile.html 文件，在此文件中对显示的文本设置不同的字体、字号和颜色，具体代码如下：

```
<body>
<font face="黑体" color="#0099FF">黑体效果</font><br>
<font size="3" color="#990000">3 号文字效果</font><br>
<font face="楷体" size="7" color="#FF6600">7 号楷体效果</font><br>
</body>
```

运行程序，效果如图 2.3 所示。



图 2.3 效果

2. 粗体

在网页中看到的粗体的文字是通过 `` 标签实现的，其语法形式如下：

```
<strong>粗体的文字</strong>
```

3. 斜体

在网页中看到的斜体字是通过 `` 标签实现的，其语法形式如下：

```
<em>斜体字</em>
```

4. 下划线

`<u>` 标签可以实现网页中文字带有下划线的效果，其语法形式如下：

```
<u>带下划线的文字</u>
```

5. 删除线

`<s>` 标签可以实现网页中文字带有删除线的效果，其语法形式如下：

```
<s>带删除线的文字</s>
```

6. 上标和下标

在开发中，网页中需要一种特殊的文字效果，即上标（上标是位于文字或数字的右上位置的标记）和下标（下标是在文字或数字的右下位置的标记，上标、下标应用也比较广泛，如 H_2O ），如在公式中经常用到。在 HTML 语言中，也可以通过标签轻松进行设置。上标可以使用 `<sup>` 标签实现，其语法形式如下：

```
<sup>上标</sup>
```

下标可以使用 `<sub>` 标签实现，其语法形式如下：

```
<sub>下标</sub>
```

实例 2-3 使用上标实现 $X^3+5X^2-3=0$ ，使用下标实现 H_2S ，具体操作步骤如下：

- (1) 打开创建的 Java Web 项目。
- (2) 创建并添加一个 HTML 文件，命名为 NewFile.html。
- (3) 打开 NewFile.html 文件，在此文件中实现上标和下标的显示，具体代码如下：

```
<body>  
X<sup>3</sup>+ 5X<sup>2</sup>-3=0<br><br>  
H<sub>2</sub>S<br>  
</body>
```



运行程序，效果如图 2.4 所示。

```
X3+5X2-3=0
H2S
```

图 2.4 效果

7. 等宽文字标签

`<code>`等宽文字标签常用于英文效果，使用该标记可以实现网页中字体的等宽效果。有些时候，使用等宽效果能够令页面显得更加整齐，其语法形式如下：

```
<code>文字</code>
```

8. 空格

一般情况下，在网页中输入文字的时候，如果在段落开始增加了空格，在使用浏览器浏览网页时往往看不到这些空格。这是因为在HTML文件中，浏览器本身会将两个句子之间的所有半角空格，仅当作一个来看待。如果需要保留空格的效果，一般需要使用全角空格符号，或者用 空格码来代替，其语法形式如下：

```
&nbsp;
```

注意：一个“ ”只代表一个半角空格，多个空格则需要多次使用这一符号。

9. 其他特殊符号

除了空格外，在网页的创作过程中，还有一些特殊的符号也需要使用代码进行代替。一般情况下，特殊符号的代码是由前缀“&”、字符名称和后缀“;”组成。使用方法与空格符号类似，具体见表 2.2。

表 2.2 特殊符号的表示

特殊符号	符号的代码
“	"
&	&
<	<
>	>
×	×
§	§
©	©
®	®
™	™



2.13 HTML 超链接标签

HTML 超链接
标签

本节将讲解与超链接标签相关的内容，主要包含文本链接、超链接网页的打开方式及链接注释等。

1. 文本链接

文本链接就是为文本加上超级链接，以达到跳转网页效果。文本链接的语法如下：

```
<a href="">文字链接</a>
```

其中，在引号里面输入要链接到的网页地址，文本链接被 `<a>` 标记对包围着，代表超链接。`href` 属性说明文字链接的地址。属性里面可以是 url 地址、图片地址或影视文件等。

实例 2-4 下面将实现文字链接，具体操作步骤如下：

- (1) 打开创建的 Java Web 项目。
- (2) 创建并添加一个 HTML 文件，命名为 NewFile.html。
- (3) 打开 NewFile.html 文件，在此文件中使用超链接标签实现单击百度网站链接跳转到百度网站中，具体代码如下：

```
<body>
<a href="https://www.baidu.com/">百度网站</a>
</body>
```

运行程序，初始效果如图 2.5 所示。单击百度网站后，会打开百度网站，如图 2.6 所示



图 2.5 初始效果



图 2.6 单击百度网站后的效果

2. 超链接网页的打开方式

`target` 属性放在 `<a>` 标签的起始标签中，表示所链接的网页在浏览器中的打开方式，该属性需要指定一个值，其值可以是 `_blank`、`_parent`、`_self` 或 `_top`，这些值代表的含义如下。

- (1) `_blank`: 在新浏览器窗口中打开网页。该值比较常用。
- (2) `_parent`: 将要链接的文件载入含有该链接框架的父框架集或父窗口中。如果含有该链接的框架不是嵌套的，则在浏览器全屏窗口中载入链接的文件，就像 `_self` 参数一样。
- (3) `_self`: 在同一框架或窗口中打开所链接的文档。此参数为默认值，通常不用指定。
- (4) `_top`: 在当前的整个浏览器窗口中打开所链接的文档，因而会删除所有框架。

3. 链接的注释

在链接标签中还有一个属性，即 `title` 属性，该属性被称为链接的注释，当鼠标指针放在文本链接上时，稍后会出现一行文字注释。



2.14 HTML 图像

要显示一个图像，需要使用到 `` 标签，其语法形式如下：

```

```

如果某些图像不能正常显示，可以使用 `` 标签的 `alt` 属性来设置图像不能正常显示时的提示信息，其基本语法形式如下：

```

```

如果要对图像的大小进行设置，可以使用 `` 标签的 `width` 属性和 `height` 属性。其中，`width` 属性表示宽度，`height` 属性表示高度。

2.15 HTML 列表

列表以条列式的模式来显示数据。在 HTML 中有 5 种列表，分别为无序列表、有序列表、自定义列表、菜单列表和目录列表。

1. 无序列表

无序列表是在每一个列表项文字之前，以符号作为分项标识的列表。创建无序列表需要用到 `` 标签，列表中的每一个列表项需要使用 `` 标签，其基本语法形式如下：

```
<ul>
  <li>第 1 项
  <li>第 2 项
  <li>第 3 项
  .....
</ul>
```

注意：无序列表的项目符号是 ●，如果想要进行修改，可以使用 `` 标签中的 `type` 属性。其基本语法形式如下：

```
<ul type=符号类型>
.....
</ul>
```

HTML 支持的符号类型有三种，见表 2.3。

表 2.3 符号类型

符号类型	表示的符号
disc	●
circle	○
square	□



2. 有序列表

在有序列表中，各个列表项使用编号来进行排列。有序列表中的项目通常都有先后顺序性，一般采用数字或字母作为顺序号。创建有序列表需要用到标签，列表中的每一个列表项需要使用标签，其基本语法形式如下：

```
<ol>
  <li>第 1 项
  <li>第 2 项
  <li>第 3 项
  .....
</ol>
```

默认情况下，有序列表的序号是数字，开发者可以通过标签中的type属性调整序号的类型，其语法形式如下：

```
<ol type=序号类型>
  .....
</ol>
```

在该语法中，序号类型可以有五种，见表 2.4。

表 2.4 有序列表的序号类型

类型	表示
1	数字 1,2,3,4,...
a	小写英文字母 a,b,c,d,...
A	大写英文字母 A,B,C,D,...
i	小写罗马数字 i , ii , iii , iv , ...
I	大写罗马数字 I , II , III , IV , ...

3. 自定义列表

自定义列表以<dl>标签开始。每个自定义列表的列表项以<dt>开始，其定义以<dd>开始。

```
<dl>
  <dt>名词 1<dd>解释 1
  <dt>名词 2<dd>解释 2
  <dt>名词 3<dd>解释 3
  .....
</dl>
```

4. 菜单列表

菜单列表主要用于设计单列的菜单列表。该列表在浏览器中的显示效果和无序列表是相同的，创建菜单列表需要用到<menu>标签，其语法形式如下：



```
<menu>
  <li>列表项 1
  <li>列表项 2
  <li>列表项 3
  .....
</menu>
```

5. 目录列表

目录列表一般用来创建多列的目录列表，它在浏览器中的显示效果与无序列表相同。创建目录列表需要用到 <dir> 标签，其语法形式如下：

```
<dir>
  <li>列表项 1
  <li>列表项 2
  <li>列表项 3
  .....
</dir>
```

2.16 HTML 表格

表格常用来对页面进行排版，表格通过三个标签来构建，分别是 <table> 表格标签、<tr> 行标签和 <td> 标签，其语法形式如下：

```
<table>
  <tr>
    <td>单元格内的文字</td>
    <td>单元格内的文字</td>
    .....
  </tr>
  <tr>
    <td>单元格内的文字</td>
    <td>单元格内的文字</td>
    .....
  </tr>
  .....
</table>
```

任务 2-1

散文鉴赏

任务描述

(1) 此任务有三个页面，分别为主页面、“春”页面及“雪”页面，介绍如下：

① 主页面：有两个超链接，一个显示“春(朱自清散文)”，另一个显示“雪(朱自清散文)”。



散文鉴赏



花在纷飞之后，却永远如粉，如沙，他们决不粘连，撒在屋上，地上，枯草上，就是这样。屋上的雪是早已就有消化了的，因为屋里居人的火的温热。别的，在晴天之下，旋风忽来，便蓬勃地奋飞，在日光中灿灿地生光，如包藏火焰的大雾，旋转而且升腾，弥漫太空，使太空旋转而且升腾地闪烁。

</body>

运行程序，首先会显示主页面，如图 2.7 所示。单击“春(朱自清散文)”链接，跳转到“春”页面，如图 2.8 所示。单击“雪(朱自清散文)”链接，跳转到“雪”页面，如图 2.9 所示。

[春\(朱自清散文\)](#)
[雪\(朱自清散文\)](#)

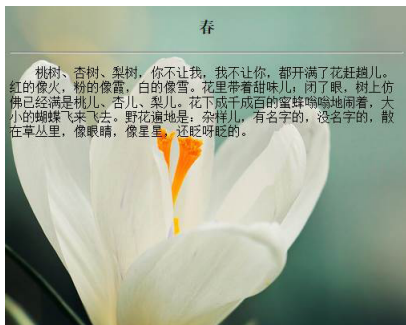


图 2.7 主页面

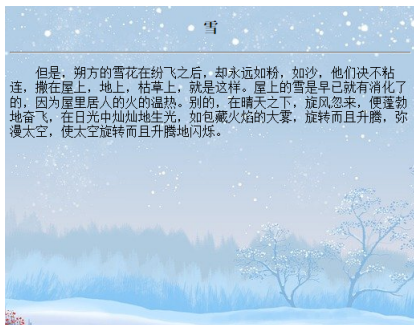


图 2.8 “春”页面



图 2.9 “雪”页面

2.2

JavaScript 基础

本节主要讲解 JavaScript 的一些基础内容，如变量、数据类型、类型转换、表达式、编程语句、函数与函数调用等。

2.2.1 编写第一个 JavaScript 程序

在正式介绍 JavaScript 基础内容之前，首先来讲解如何编写 JavaScript 程序，具体的操作步骤如下：

- (1) 创建 Java Web 项目。
- (2) 在项目中添加一个 HTML 文件，名称为 NewFile.html。
- (3) 打开 NewFile.html 文件，编写代码，弹出一个对话框，代码如下：

```
<body>
<script>
alert("欢迎使用JavaScript!");
</script>
</body>
```

此代码使用了 <script> 标签，通过该标签可以将脚本代码嵌入 HTML 文档中，如 JavaScript。在 <script> 和 </script> 之间的代码就是 JavaScript，该代码实现了弹出一个对话框的功能。

运行程序，会弹出一个对话框，如图 2.10 所示。

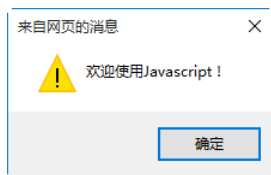


图 2.10 对话框

注意：为了对 JavaScript 代码更好地进行管理，可以将该代码单独存放在 JavaScript 文件中。JavaScript 文件的创建方法如下：右击项目名，在弹出的菜单中选择 New|JavaScript File 命令，弹出 New JavaScript file 对话框，输入文件名称，单击 Finish 按钮即创建完成。本书创建一个 MyJavaScript 文件，此文件的后缀名为 .js。打开 MyJavaScript.js 文件（可以选择文本编辑器的方式打开），在此文件中添加 JavaScript 代码，代码如下：

```
alert ("欢迎使用 JS");
```

返回 HTML 文件中，使用 <script> 标签来引用 MyJavaScript.js 文件，代码如下：

```
<body>  
<script type="text/javascript" src="MyJavaScript.js">  
</script>  
</body>
```

运行程序，还是会弹出一个对话框，此时对话框中的内容为“欢迎使用 JS”。

2.2.2 变量

变量其实就是一个名称，用来存储计算结果或表示某个数据类型的值。这些数据可以是一个数字，也可以是字符或某个对象。在 JavaScript 中，定义变量的方式有两种，分别为赋值时定义和使用 var 定义。

1. 赋值时定义

在为变量进行第一次赋值时定义变量，其形式如下：

```
变量名 = 数据；
```

2. 使用 var 定义

使用关键字 var 定义变量的形式如下：

```
var 变量名；
```

或是

```
var 变量名 = 数据；
```

在 JavaScript 中，对于变量的命名需要遵循以下几个规定：

- (1) 变量名只能包含字母、数字及下划线，其他字符都是非法的。
- (2) 变量名只能以字母开头。



(3) 变量名不能采用JavaScript的保留字，如var等。

注意：一般建议使用第二种方式去定义变量，因为使用第一种方式定义的变量将不会在列表框中显示。

2.2.3 基本数据类型

JavaScript支持的基本数据类型有6种，见表2.5。

表 2.5 数据类型

数据类型	说明
Number(数字)	可以是一个整数，也可以是一个浮点数
String(字符串)	一个字符串变量
Boolean(布尔)	true 或 false
Null(空)	空值
Undefined(空)	
Object(空)	数据元素属性及方法的集合

当创建字符串变量时，有些字符是不能显示的，如换行符、单引号等，此时就需要用到转义字符。JavaScript支持的转义字符见表2.6。

表 2.6 转义字符

转义字符	说明
\'	插入单引号
\"	插入双引号
\b	插入退格
\f	插入换页符
\n	插入换行符
\t	插入指表符
\r	插入回车符

2.2.4 表达式

运算符是一种告诉编译器执行特定的数学或逻辑操作的符号。在JavaScript中，支持的运算符见表2.7。



表 2.7 运算符列表

运算符	说明
算术运算符	采用数值常量及变量作为操作数，并返回数值
复杂运算符	进行赋值操作
位运算符	在二进制形式上进行运算，返回一个数字
逻辑运算符	返回布尔值
比较运算符	比较操作数并返回一个布尔值
字符串运算符	以字符串为操作数，并返回一个字符串
赋值运算符	将右操作数赋给一个左操作数

使用数据和运算符组合的合法序列就被称为表达式。根据运算符的种类，JavaScript 的表达式可以分为赋值表达式、算术表达式、逻辑表达式、条件表达式等。其中，最为特殊的一种表达式就是条件表达式，其格式如下：

```
逻辑表达式? 表达式 1: 表达式 2
```

当逻辑表达式返回 true 时，执行表达式 1，否则执行表达式 2。

2.2.5 类型转换

在使用运算符进行运算时，常常会遇到两个操作数类型不同的情况。此时，就需要用到类型转换。

1. 将任意类型的值转换为字符串

任何类型的变量都有一个 toString() 方法，通过这个方法，可以将相应类型的值转换成字符串，如以下的代码，此代码是将数字类型的值转换为字符串。

```
var i = 123456;
var s = i.toString();
alert(s);
```

2. 字符串转换为数字类型

JavaScript 中提供了两个函数用来将字符串转换成数字，即 parseInt() 和 parseFloat()，其中，parseInt() 可以将字符串转换成整型值，parseFloat() 可以将字符串转换成浮点值。

3. 强制类型转换

JavaScript 支持三种强制类型转换，分别介绍如下：

- (1) Boolean (value): 把 value 中的值转换成 Boolean 类型。
- (2) Number (value): 把 value 中的值转换成数字 (整数或浮点数)。
- (3) String (value): 把 value 中的值转换成字符串。



2.2.6 编程语句

语句就是以分号结尾的代码。在 2.2.5 中提到的代码就属于一条一条的语句。当多条语句中出现比较复杂的代码时，可以采用语句块的形式，将语句分开。语句块是指被一组“{}”符号包含的语句。各个语句之间采用“;”。JavaScript 主要支持的三种编程语句为条件语句、循环语句和终止语句。

1. 条件语句

条件语句是判断当一个逻辑表达式为真时，执行一个动作；当其不为真时，执行另一个动作。条件语句采用的是 if...else... 的模式，其语法格式如下：

```
if(逻辑表达式) 表达式 1;
```

或

```
if(逻辑表达式){
    表达式 1;
}else{
    表达式 2;
}
```

2. 循环语句

循环语句是重复执行某个动作，直到循环条件不为真 (false) 时为止。循环语句主要有两种表现形式，一个是 for 语句，另一个是 while 语句。在 for 语句中，参数有初始值、循环条件及执行循环的次数；while 语句的参数只有循环条件。for 语句的语法格式如下：

```
for(var i=0;i<num;i++){
    表达式;
}
```

While 语句的语法格式如下：

```
While(逻辑表达式){
    表达式;
}
```

3. 终止语句

在循环语句内，常常并不需要执行所有的循环操作。这时就可以使用终止语句来实现该操作。JavaScript 支持两种终止语句，分别为 break 语句和 continue 语句。其中，break 语句终止循环后，跳出此循环语句，执行整个循环体后面的操作；continue 语句并不离开整个循环体，只是跳出本次循环，开始下一次循环操作。break 语句的语法格式如下：

```
for(var i=0;i<num;i++){
    if(逻辑表达式) break;
    表达式;
```



```
}
```

continue 语句的语法格式如下:

```
for(var i=0;i<num;i++){  
    if(逻辑表达式) continue;  
    表达式;  
}
```

2.2.7 函数与函数调用

在JavaScript中,函数需要使用关键字function来声明,函数的基本语法如下:

```
function funName(形参列表)  
{  
    statements  
}
```

函数可以加多个参数,中间使用逗号(,)分隔。例如,以下的代码定义了一个sayHello()的函数:

```
function greet(sName)  
{  
    alert("你好 " + sName);  
}
```

如果要调用函数,使用以下形式:

```
funName(实参列表)
```

例如,以下的代码是调用了sayHello()函数:

```
sayHello
```

2.2.8 JavaScript 对象

在JavaScript中,对象大致分为两种,一种是数据类型对象,如字符串类型对象、数组对象等;还有一种是控件类型对象(对页面上控件的封装),如按钮对象、复选框对象等。不论是数据类型对象还是控件类型对象都包含属性、方法及事件三部分。根据对象分类的不同,属性也是不同的。对于数据类型对象,其属性可以是字符串的长度,可以是一个数字等;对于控件类型对象,其属性可以是一个文本框的宽度,可以是一个图层的颜色等。本小节介绍JavaScript内置数据类型对象,包含String(字符串)对象、Array(数组)对象、Math(数学)对象、Date(日期)对象。

1. String(字符串)对象

字符串对象的声明可以用单引号(')完成,也可以用双引号("")完成,其形式如下:



```
var 变量名 = new String("字符串");
```

或是:

```
var 变量名 = "字符串";
```

字符串对象的常用属性为 `length`，主要返回字符串的长度。字符串对象的常用方法见表 2.8。

表 2.8 字符串对象的常用方法

方法名	说明
<code>charAt(offset)</code>	返回字符串中第 <code>(offset+1)</code> 个字符
<code>charCodeAt(offset)</code>	返回字符串中第 <code>(offset+1)</code> 个字符的 ASCII 码
<code>indexOf(string)</code>	返回字符串中 <code>string</code> 出现的位置，如果字符串中不存在 <code>string</code> ，则返回 <code>-1</code>
<code>lastIndexOf(string)</code>	返回字符串中 <code>string</code> 最后出现的位置，如果字符串中不存在 <code>string</code> ，则返回 <code>-1</code>
<code>split(string)</code>	将字符串拆分为一个数组，拆分的原则是以 <code>string</code> 为分割
<code>substring(startpos, endpos)</code>	返回字符串的一个子字符串。该字符串是从主字符串的 <code>startpos</code> 位置到 <code>endpos</code> 位置之间截取的
<code>toLowerCase()</code>	将字符串全部小写返回
<code>toUpperCase()</code>	将字符串全部大写返回

2. Array (数组) 对象

数组对象定义的是一个对象的集合，这些对象可以是不同类型的。数组对象的定义形式如下：

```
var 变量名 = new Array();
```

或

```
var 变量名 = new Array(<元素对象 1>, <元素对象 2>, ...);
```

数组对象的常用属性为 `length`，主要返回数组中对象的个数。数组对象的常用方法见表 2.9。

表 2.9 数组对象的常用方法

方法名	说明
<code>join(string)</code>	将数组中各对象通过 <code>string</code> 连接起来组成一个字符串并返回
<code>reverse()</code>	将数组中的元素对象倒排
<code>slice(startpos, endpos)</code>	返回数组的一个子集，此子数组是从主数组的 <code>startpos</code> 位置到 <code>endpos</code> 位置之间截取的



3. Math(数学)对象

数学对象通常是用来计算页面上控件的位置等数据。数学对象的属性见表 2.10。

表 2.10 数学对象的属性

属性名	说明
E	常数e
LN2	2 的自然对数
LN10	10 的自然对数
LOG2E	以 2 为底的e的对数
LOG10E	以 10 为底的e的对数
PI	圆周率
SQRT1_2	1/2 的平方根
SQRT2	2 的平方根

数学对象的常用方法见表 2.11。

表 2.11 数学对象的常用方法

方法名	说明
abs(x)	X 的绝对值
acos(x)	X 的反余弦值
asin(x)	X 的反正弦值
exp(x)	e 的 X 次幂
floor(x)	小于或等于 X 的最大整数
max(a,b)	最大值
min(a,b)	最小值
pow(a,b)	a 的 b 次幂
random()	0 到 1 之间的一个随机数
sqrt(x)	X 的平方根
sin(x)	X 的正弦值
cos(x)	X 的余弦值

如果要调用表 2.11 中的方法，可以使用以下的形式：

```
Math.方法名(实参);
```



4. Date(日期)对象

日期对象主要是为了定义时间。这个时间可以精确到毫秒级。定义一个日期对象的语法形式如下:

```
var 变量名 = new Date;
```

日期对象的常用方法见表 2.12。

表 2.12 日期对象的常用方法

方法名	说明
getFullYear()	获取年份
getMonth()	获取月份
getDate()	获取日期
getDay()	获取星期, 0 为星期天
getHours()	获取小时, 24 小时制
getMinutes()	获取分钟
getSeconds()	获取秒数
getMilliseconds()	获取毫秒数
setYear(x)	设置年份
setMonth(x)	设置月份
setDate(x)	设置日期
setDay(x)	设置星期, 0 为星期天
setHours(x)	设置小时, 24 小时制
setMinutes(x)	设置分钟
setSeconds(x)	设置秒数
setMilliseconds(x)	设置毫秒数

2.2.9 DOM(文档)对象

DOM(文档)对象是对页面上的各个元素的封装。本小节主要介绍几个常用的对象。

1. document 对象

实际的文档对象有很多种, 以下主要介绍 document 对象及其子对象。document 对象的子对象见表 2.13。



表 2.13 document对象的子对象

对象名称	说明
anchors [], links [], Link	链接对象
applets []	applet小应用程序对象
embeds []	插件对象
forms [], Form	表单对象
images [], Image	图片对象

其中，表单对象的子对象见表 2.14。

表 2.14 表单对象的子对象

对象名称	说明
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	密码输入框对象
Radio	单选框对象
Reset	重置按钮对象
Select	下拉框对象
Submit	提交按钮对象
Text	文本输入框对象
Textarea	多行文本输入框对象

因为文档对象具有层级附属关系，所以当引用一个子对象时，需要将其父对象一起列出。例如，以下的代码是一个表单。

```
<form id="my" name="my">  
<input type="text" name="mytext"/>  
</form>
```

当需要引用mytext文本框对象时，需要用到以下的代码：

```
<script>  
var mytextobj = document.forms["my"]."mytext";  
</script>
```

在JavaScript中，使用大写字母开头的对象名，可以直接使用id或name名来引用。document对象封装的是当前窗口或指定窗口的文档，它包括了文档页面的<head></head>和



<body></body>之间的内容。document对象的主要属性见表 2.15。

表 2.15 document对象的主要属性

属性名	说明
Cookie	Cookie 信息
lastModified	文档最后修改时间
title	<title>...</title> 节点之间的内容
fgColor	<body> 标签中 text 属性表示的文本颜色
bgColor	<body> 标签中 bgcolor 属性表示的背景颜色
linkColor	<body> 标签中 link 属性表示的链接颜色

document对象的主要方法见表 2.16。

表 2.16 document对象的主要方法

方法	说明
write(string)	向文档中写入内容
writeln(string)	向文档中写入内容，完成此方法后文档将另起一行
clear()	清空文档
getElementById (string)	按照 string 查找文档中相应的子对象

2. Form(表单)对象

Form对象是document对象的子对象，其属性见表 2.17。

表 2.17 Form对象的属性

属性名	说明
name	表单的名称
action	表单提交的地址
method	表单提交的方法
target	表单提交后返回窗口

Form对象的常用方法见表 2.18。

表 2.18 Form对象的常用方法

方法名	说明
reset()	重置表单
submit()	提交表单

如果要引用Form对象，可以直接通过名称来完成，其形式如下：



```
document.<表单名>
```

3. Button(按钮)对象

Button对象的属性见表 2.19。

表 2.19 Button对象的属性

属性名	说明
name	元素的名称
value	元素的值
form	包含本元素的表单对象

Button对象的常用方法见表 2.20。

表 2.20 Button对象的常用方法

方法	说明
blur()	从对象中移走焦点
focus()	让对象获得焦点
click()	模拟鼠标单击事件

Button对象的定义形式如下：

```
<input type="button">
```

Button对象由大写字母开头，可以直接通过其名称来引用，引用Button对象的语法形式如下：

```
文档对象.表单对象名称.按钮名称
```

其中，按钮名称为按钮中的“name=“...””属性的值。

4. Checkbox(复选框)对象

Checkbox对象的属性见表 2.21。

表 2.21 Checkbox对象的属性

属性名	说明
name	元素的名称
value	元素的值
form	包含本元素的表单对象
checked	设定该复选框对象是否被选中

Checkbox对象的常用方法见表 2.22。



表 2.22 Checkbox对象的常用方法

方法	说明
blur()	从对象中移走焦点
focus()	让对象获得焦点
click()	模拟鼠标单击事件

Checkbox对象的定义形式如下：

```
<input type="checkbox">
```

它可以实现在列表中同时选择多个项目。因为Checkbox对象是大写开头，所以可以直接通过其名称来引用，其形式如下：

```
文档对象.表单对象名称.复选框名称
```

其中，复选框名称为按钮中的“name=“...””属性的值。

5. Password(密码输入框)对象

Password对象的属性见表 2.23。

表 2.23 Password对象的属性

属性名	说明
name	元素的名称
value	元素的值
form	包含本元素的表单对象

Password对象的常用方法见表 2.24。

表 2.24 Password对象的常用方法

方法	说明
blur()	从对象中移走焦点
focus()	让对象获得焦点
select()	选中密码输入区里的全部文本

Password对象的定义形式如下：

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

因为Password对象是大写字母开头，所以可以直接通过其名称来引用，其形式如下：

```
文档对象.表单对象名称.密码输入框名称
```

其中，密码输入框名称为按钮中的“name=“...””属性的值。

6. Text(文本输入框)对象

Text对象的属性见表 2.25。



表 2.25 Text对象的属性

属性名	说明
name	元素的名称
value	元素的值
form	包含本元素的表单对象

Text对象的常用方法见表 2.26。

表 2.26 Text对象的常用方法

方法	说明
blur()	从对象中移走焦点
focus()	让对象获得焦点
select ()	选中文本输入区里的全部文本

Text对象的定义形式如下：

```
<input type=" text ">
```

因为文本输入框对象是大写字母开头，所以可以直接通过其名称来引用，其形式如下：

```
文档对象.表单对象名称.文本输入框名称
```

其中，文本框名称为按钮中的“name=“...””属性的值。



任务 2-2

使用按钮改变背景颜色

任务描述

(1)显示一个页面，在此页面中会有一个div元素和三个按钮控件，分别为“红色”“黄色”“蓝色”。

(2)单击“红色”按钮，div元素的背景颜色会变为红色。单击“黄色”按钮，div元素的背景颜色会变为黄色。单击“蓝色”按钮，div元素的背景颜色会变为蓝色。

任务实施

1. 创建项目

创建Java Web项目，项目名称为BackgroundDemo。

2. 添加HTML文件

在项目中添加一个HTML文件，名称为NewFile.html。

3. 修改NewFile.html文件的代码

在NewFile.html文件中实现对主页面的布局，以及单击按钮触发的改变背景颜色的功能，代码如下：

```
<body>
```



使用按钮改变背景颜色



```

<script>
//单击按钮背景变为红色
function buttonRedClick(){
    document.getElementById("P1").style.background='red';
}
//单击按钮背景变为黄色
function buttonYellowClick(){
    document.getElementById("P1").style.background='yellow';
}
//单击按钮背景变为蓝色
function buttonBlueClick(){
    document.getElementById("P1").style.background='blue';
}
</script>
<div id="P1" style="height:100px; width:155px;" >单击对应按钮改变段落背景色</div>
<input id="redButton" type="button" value="红色"
onclick="buttonRedClick()">
<input id="yellowButton" type="button" value="黄色" onclick="buttonYellowClick();">
<input id="blueButton" type="button" value="蓝色"
onclick="buttonBlueClick();">
</body>

```

运行程序，初始效果如图 2.11 所示。当单击“黄色”按钮后，div 元素的背景颜色会变为黄色，如图 2.12 所示。

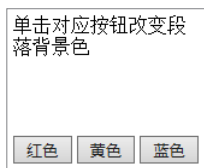


图 2.11 初始效果

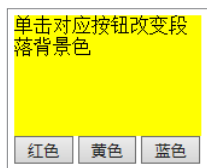


图 2.12 改变背景颜色



任务 2-3

图片随机展示

任务描述

- (1) 显示一个图片展示页面，此页面中有一个按钮和一个图片。
- (2) 单击按钮，图片会自动进行展示。

任务实施

1. 创建项目

创建 Java Web 项目，项目名称为 ShowImageDemo。



图片随机展示



2. 添加图像

将 1.jpg、2.jpg、3.jpg、4.jpg、5.jpg、6.jpg 图像文件分别添加到项目中。

3. 添加 HTML 文件

在项目中添加一个 HTML 文件，名称为 NewFile.html。

4. 修改 NewFile.html 文件的代码

在 NewFile.html 文件中实现对页面的布局及单击按钮播放图像的功能，代码如下：

```
<body>
<script>
function showP()                                //单击切换图片函数
{
    var Num=Math.floor((Math.random()*6)+1);
    var fImg = document.getElementById("timg");
                                                //通过id获取图片标签
    imgSrc = Num + ".jpg" ;                    //根据函数实参修改变量中的路
径
    fImg.src=imgSrc;                            //设置图片表的src属性值

    setInterval(function()                    //定时器
    {
        showP();                                //调用轮转函数
    },6000);
}
</script>
<div id="Box">
<!--焦点图片-->
<br />
<button onClick="showP()">单击随机展示图片</button>
</div>
</body>
```

运行程序，初始效果如图 2.13 所示。单击“单击随机展示图片”按钮，图片实现自动播放。

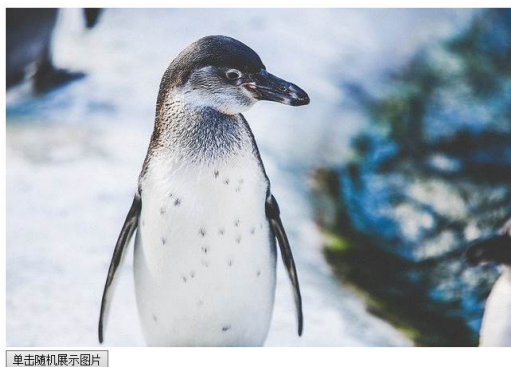


图 2.13 初始效果



八进制或十六进制转换字符串。

被转换的是八进制或十六进制数，使用 toString() 方法仍然以十进制输出这些数，示例代码如下：

```
var om = 0245;           // 定义八进制整数
var hx = 0xA1;          // 定义十六进制整数
var som = om.toString(); // 八进制转换为字符串，值为 165
var shx = hx.toString(); // 十六进制转换为字符串，值为 161
alert(som);             // 弹出内容为 165 的对话框
alert(shx);             // 弹出内容为 161 的对话框
```

如果想直接获得二进制、八进制和十六进制的变量值，可以使用如下代码：

```
var b = 12;
var hx = 0xA1;
var sb = om.toString(2); // 值为 1100
var shx = hx.toString(16); // 值为 a1
alert(sb);               // 弹出内容为 1100 的对话框
alert(shx);              // 弹出内容为 a1 的对话框
```



一、填空题

1. 在网页中，定义标题使用的标签是_____。
2. 在网页中，粗体字是通过_____标签实现。

二、选择题

1. 下列选项中，()表示的不是按钮。
A. type="submit" B. type="reset" C. type="image" D. type="button"
2. 下列选项中，()不是文本的标签属性。
A. size B. align C. Color D. face

三、判断题

1. HTML 语言可以直接描述图像上的像素。 ()
2. 在 HTML 中，标签 的 size 属性最大值是 7。 ()

四、操作题

实现单击按钮改变字体大小。

