

## 第 5 章 JSP 基础知识

JSP 页面主要由 HTML、CSS、Java 代码段(脚本元素)、注释、JSP 指令和 JSP 动作等构成。本章主要讲解 JSP 页面的基本语法构成。

本章主要内容如下所示。

- (1) JSP 页面的基本结构。
- (2) JSP 的注释。
- (3) JSP 的脚本元素。
- (4) JSP 的指令。
- (5) JSP 的动作。

### 5.1 JSP 页面的基本结构

一个 JSP 页面由以下基本元素组成。

- (1) HTML 标签。
- (2) CSS。
- (3) 变量和方法。
- (4) Java 代码段。
- (5) JSP 动作和指令。
- (6) 其他脚本元素(如 JavaScript)等。

本节通过一个 JSP 实例,了解 JSP 页面的基本语法构成。具体的 JSP 基本语法部分将在以后的章节里详细介绍。

**【例 5-1】** JSP 页面的基本结构实例(pageStructure.jsp)。

```
1. <%@page contentType="text/html" pageEncoding="UTF-8"%>
2. <html>
3.     <head>
4.         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5.         <title>JSP 页面的基本结构实例</title>
6.     </head>
7.     <body>
8.         <%!int sum=0;
9.             int x=1;
10.             %>
11.         <%
12.             while (x <=10)
13.             {
14.                 sum+=x;
15.                 ++x;
16.             }
```

```

17.         %>
18.         <p>1 加到 10 的结果是:<%=sum%></p>
19.         <p>现在的时间是:<%=new java.util.Date()%></p>
20.     </body>
21. </html>

```

本程序的功能是计算 1 加到 10 的结果并将结果在页面第一行中显示,在第二行显示当前系统时间,如图 5-1 所示。

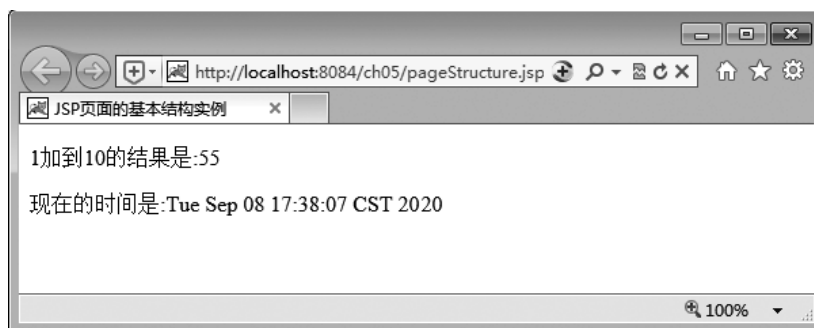


图 5-1 页面运行效果

从例 5-1 中可以看到,JSP 页面除了比普通的 HTML 页面多一些 Java 代码、指令和动作外,两者的基本结构相似。实际上,JSP 基本元素是嵌入在 HTML 页面中的,为了和 HTML 的标签有所区别,JSP 标记都以“<%”或“<jsp”开头,以“%>”或“>”结尾。下面对该 JSP 文件进行详细介绍。

第 1 行是 JSP 的 page 指令,它描述 JSP 文件转换成 JSP 服务器所能执行的 Java 代码的控制信息,如 JSP 页面所使用的语言、对处理内容是否使用缓存、是否线程安全、错误页面处理、指定内容类型、指定页面编码方式等。例如,contentType="text/html"用于指定内容类型,pageEncoding="UTF-8"用来指定页面的编码方式。

第 2~7 行是一些 HTML 常用的标签,在第 3 章中已介绍,这里不再赘述。

第 8~10 行是 JSP 中的声明。JSP 页面中用到的变量和方法与 Java 程序中用到的变量和方法的声明是相同的,不过在 JSP 页面中声明以“<%!”或者“<%”开头,以“%>”结尾。本例中对两个整型变量的声明并初始化,也可以写成“<%!int sum=0;int x = 1; %>”。

第 11~17 行是 JSP 中的程序代码,即 JSP 中的脚本。JSP 的程序代码封装了 JSP 页面的业务处理逻辑——Java 代码程序,以“<%”开头,以“%>”结尾。

第 12~16 行是一段标准的 Java 程序,其功能是对 1 加到 10 的计算。

第 18 行,“<p>1 加到 10 的结果是:<%=sum%></p>”中的“<%=sum%>”是表达式,在 JSP 中表达式以“<%=”开头,以“%>”结尾。本行语句用于输出 1 加到 10 的结果。

第 19 行,“<p>现在的时间是:<%=new java.util.Date()%></p>”中的“<%=new java.util.Date()%>”是使用表达式以及 Java 类库中的 Date 类获取系统的当前时间。

第 20、21 行是 HTML 的基本标签。

通过上面典型的 JSP 页面可以看出,JSP 页面就是在 HTML 或者 XML 代码中嵌入 Java 语法或者 JSP 元素,从而实现系统的业务功能,这一点读者将会在以后的学习中更有

深刻的体会。

**想一想：**JSP 页面设计要遵循既定的语法规则和开发规范。你对软件开发规范的重要性有什么认识？作为一名软件工程师，你认为应该遵循哪些工程职业道德规范？

## 5.2 JSP 的 3 种常用注释



JSP 常用  
注释

程序中注释的作用是为了提高程序的可读性、可维护性和可扩展性，所以一个 Java Web 项目中需要各种各样的注释。在 JSP 中注释有 3 种类型：隐藏注释、HTML 注释和 Java 语言注释。下面分别介绍这 3 种注释的使用。

### 5.2.1 隐藏注释及其应用实例

隐藏注释是 JSP 的标准注释，写在 JSP 程序中，用于描述和说明 JSP 程序代码，在发布 JSP 网页时完全被忽略，也不会输送到客户浏览器上。当希望隐藏 JSP 程序的注释时是很有用的。

其语法格式为

```
<%-- 注释语句 --%>
```

注释语句为要添加的注释内容。

**【例 5-2】** 隐藏注释实例(hideNotes.jsp)。

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>隐藏注释实例</title>
    </head>
    <body>
        <h3>本页面是演示隐藏注释的功能，在 JSP 页面中的隐藏注释不会发布到客户端！
        </h3>
        <hr>
        <%-- 这一行注释在发布网页时不会被看到 --%>
        <hr>
    </body>
</html>
```

hideNotes.jsp 运行效果如图 5-2 所示。在发布网页时看不到注释，在客户端浏览器源文件中也看不到注释。在浏览器中查看源文件的方法：单击“查看”→“源文件”命令。

### 5.2.2 HTML 注释及其应用实例

HTML 注释在发布网页时可以在浏览器源文件窗口中看到，即注释的内容会被输送到客户端的浏览器中。该注释中也可以使用 JSP 的表达式。

其语法格式为

```
<!-- 注释语句 [ <%=表达式 %> ] -->
```



图 5-2 hideNotes.jsp 的运行效果以及源文件

其中,注释语句是文字说明,表达式为 JSP 表达式。

### 【例 5-3】 HTML 注释实例(HTMLNotes.jsp)。

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>HTML 注释实例</title>
    </head>
    <body>
        <h3>本页面是演示 HTML 注释的功能,在 JSP 页面中的 HTML 注释会在客户端源文件中看到,但是不会发布到页面上!
        </h3>
        <hr>
        <!--这一行注释在发布网页时不会被看到,在源文件中可以看到<%=new java.util.
            Date() %>-->
        <hr>
    </body>
</html>
```

HTMLNotes.jsp 的运行效果如图 5-3 所示。在发布网页时看不到注释,但在源文件中可以看到注释,而且表达式是动态的,即根据表达式的值输出结果。

### 5.2.3 Java 注释及其应用实例

在 JSP 程序中,也可遵循 Java 语言本身的注释规则对代码进行注释,这样的注释和隐藏注释相似,在发布网页时完全忽略,在浏览器的源文件窗口中也看不到注释内容。



图 5-3 HTMLNotes.jsp 的运行效果以及源文件

其语法格式为

```
<% /* 注释语句 */ %>
```

或者

```
<%//注释语句%>
```

其中,注释语句为要添加的注释文本。

**【例 5-4】** Java 注释实例(JavaNotes.jsp)。

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Java 注释实例</title>
  </head>
  <body>
    <h3>本页面是演示 Java 注释的功能,在 JSP 页面中的 Java 注释不会发布到客户端!
    </h3>
    <hr>
    <%//这一行注释在发布网页时不会被看到,在源文件中也看不到%>
    <hr>
  </body>
</html>
```

JavaNotes.jsp 的运行效果如图 5-4 所示。在发布网页时看不到注释,在源文件中也看不到注释。



图 5-4 JavaNotes.jsp 的运行效果以及源文件



JSP 常用脚本元素

## 5.3 JSP 常用脚本元素

在 JSP 页面中,经常使用一些变量、方法、表达式以及脚本,下面分别介绍这些基本元素的使用。

### 5.3.1 变量和方法的声明及其应用实例

在 JSP 页面中可以声明一个或者多个合法的变量和方法,声明后的变量和方法可以在本 JSP 页面的任何位置使用,并将在 JSP 页面初始化时被初始化。

语法格式为

```
<%!语句 1; ...; [ 语句 n; ] %>
```

其中,语句主要用来声明变量、方法。

在声明变量和方法时,需要注意以下几点。

- (1) 声明以“<%!”或者“<%”开头,以“%>”结尾。
- (2) 变量声明必须以“;”结尾。
- (3) 变量和方法的命名规则与 Java 中变量和方法的命名规则相同。
- (4) 可以直接使用在<%@ page%>中被包含进来的已经声明的变量和方法,不需要重新声明。
- (5) 一个声明仅在一个页面中有效。如果想在每个页面都用到某些声明,最好把它们写成一个单独的文件,然后用<%@ include%>指令或<jsp:include>动作包含进来。

例如:

```
<%! int i=0;%>
```

```
<%! int x , y, z, a, b, c, sum; %>
<%! String str="北京!"; %>
<%! Date date=new java.util.Date(); %>
```

**【例 5-5】** 变量和方法的声明实例(declare.jsp)。

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>变量和方法的声明实例</title>
    </head>
    <body>
        <%!
            String str="JSP 技术带你走进动态网页开发时代!";           //变量声明
        %>
        <%!
            String print() {                                           //方法声明
                return str;
            }
        %>
        <%=print()%>
    </body>
</html>
```

declare.jsp 的运行效果如图 5-5 所示。

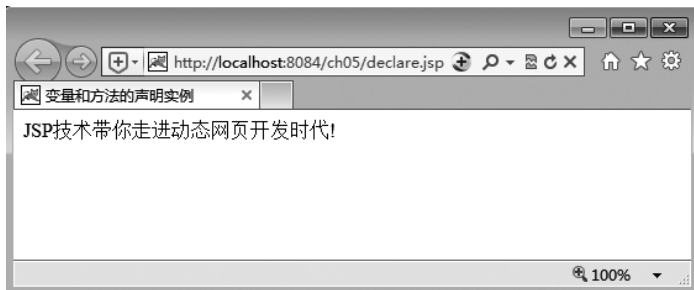


图 5-5 declare.jsp 的运行效果

在 declare.jsp 中声明了一个字符串变量 str 和一个方法 print(), 该方法用来返回字符串变量 str 的值。

### 5.3.2 表达式和脚本及其应用实例

#### 1. 表达式及其应用实例

JSP 中的表达式是由变量、常量和运算符组成, 可以将数据转换成一个字符串并直接在网页上输出。

表达式的语法格式为

```
<%=表达式%>
```

JSP 的表达式中没有分号。

JSP 的表达式常用在以下几种情况。

- (1) 向页面输出内容。
- (2) 生成动态的链接地址。
- (3) 动态指定 form 表单处理页面。

**注意：**在“<%”与“=”之间不要有空格。

**【例 5-6】** 表达式实例(expression.jsp)。

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>表达式实例</title>
    </head>
    <body>
        <%!
            String name="清华大学出版社";
            String urlAddress="www.tup.tsinghua.edu.cn";
            String page="www.sohu.com";

            %>
        <br>
        用户名: <%=name%>
        <br>
        <a href="<%=urlAddress%>">清华大学出版社网站</a>
        <br>
        <form action="<%=page%>">
        </form>
    </body>
</html>
```

本例演示了表达式的几种常用方式,expression.jsp 的运行效果如图 5-6 所示。

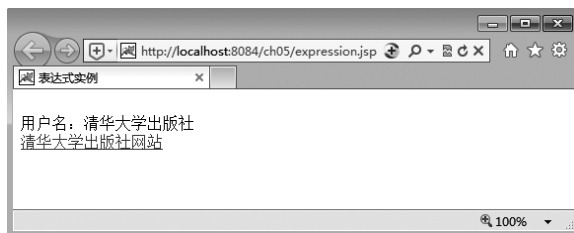


图 5-6 expression.jsp 的运行效果

## 2. 脚本及其应用实例

JSP 脚本即 Script,也就是 JSP 中的代码部分,是一段 Java 代码,几乎可以使用任何 Java 语法,它是在请求期间执行,可以使用 JSP 页面所定义的变量、方法、表达式或者 JavaBean。脚本定义的变量和方法在当前整个页面内有效,但不会被其他线程共享,用户对该变量的作用不会影响其他用户,当变量所在页面关闭时该变量就会被销毁。

脚本的语法格式为

<% 脚本语句 %>



**【例 5-7】** 脚本实例(scriptlet.jsp)。

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>脚本实例</title>
  </head>
  <body>
    <%!
      int x=6;
    %>
    <table>
      <%
        if(x==1) {
      %>
      <tr>
        <td>欢迎登录,您的权限是管理员!</td>
      </tr>
      <%
        }
        else{
      %>
      <tr>
        <td>欢迎登录,您的权限是普通用户!</td>
      </tr>
      <%
        }
      %>
    </table>
  </body>
</html>
```

scriptlet.jsp 的运行效果如图 5-7 所示。

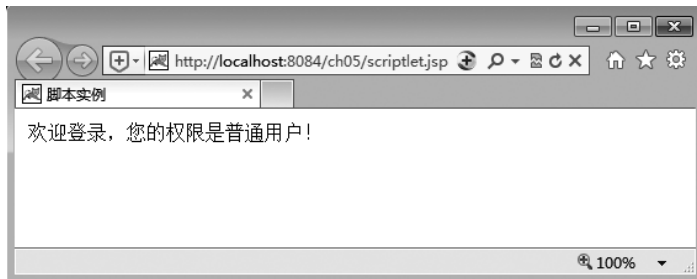


图 5-7 scriptlet.jsp 的运行效果

JSP 中大部分功能的实现可以通过 JSP 脚本实现。使用脚本程序比较灵活,它所实现的功能是 JSP 表达式无法实现的,所以脚本在 JSP 中非常重要。有关 JSP 脚本的内容在以后的章节中涉及很多,读者可以在今后的开发中不断学习和应用。

JSP 常用  
指令

## 5.4 JSP 常用指令

指令(Directive)描述 JSP 文件转换成 JSP 服务器所能执行的 Java 代码的控制信息,用于指定整个 JSP 页面的相关信息,并设置 JSP 页面的相关属性。

常用的 JSP 指令有 3 种: page 指令、include 指令和 taglib 指令。

### 5.4.1 page 指令及其应用实例

page 指令用来定义 JSP 文件中的全局属性,它描述了与页面相关的一些信息,其作用域为它所在 JSP 页面和其包含的文件。页面指令一般位于 JSP 页面的顶端,但是可以放在 JSP 页面的任何位置,无论把`<%@ page %>`指令放在 JSP 文件的哪个地方,它的作用范围都是整个 JSP 页面。为了 JSP 程序的可读性以及养成良好的编程习惯,最好把它放在 JSP 文件的顶部。

在同一个 JSP 页面中可以有多多个 page 指令。在使用多个 page 指令时,其属性除 import 属性外只能使用一次。

page 指令的语法格式如下:

```
<%@page
    [ language="java" ]
    [ extends="package.class" ]
    [ import="{package.class|package.* }, ..." ]
    [ session="true|false" ]
    [ buffer="none|8KB|sizeKB" ]
    [ autoFlush="true|false" ]
    [ isThreadSafe="true|false" ]
    [ info="text" ]
    [ errorPage="relativeURL" ]
    [ contentType="mimeType [ ;charset=characterSet ]" ]
    [ pageEncoding=" pageEncoding" ]
    [ isErrorPage="true | false" ]
%>
```

下面分别对这些属性的含义和用法进行介绍。

#### 1. language 属性

**【功能说明】** language 属性用于指定 JSP 页面中使用的脚本语言,其默认值为 Java。根据 JSP 2.0 规范,目前只可以使用 Java 语言。

例如:

```
<%@page language="java" %>
```

如果 language 属性指定了其他的脚本语言,将会产生异常。

#### 2. extends 属性

**【功能说明】** extends 属性用于指定 JSP 编译器父类的完整限定名,此 JSP 页面产生的 Servlet 将由该父类扩展而来。

例如: