



第4章 JSP语法

- ◆ 4.1 第一个JSP页面
- ◆ 4.2 注释
- ◆ 4.3 JSP表达式
- ◆ 4.4 JSP程序段
- ◆ 4.5 JSP声明
- ◆ 4.6 URL传值
- ◆ 4.7 JSP指令和动作



4.1 第一个 JSP 页面



1、JSP概述

- ◆ JSP的全名为Java Server Pages，即java服务器页面，是由Sun公司倡导、许多公司参与一起建立的一种**动态网页技术标准**。
- ◆ JSP源文件是在传统网页的**HTML**代码中**插入Java脚本**和**JSP标记**而构成的文件，其扩展名为**.jsp**。
- ◆ 用JSP开发的Web应用是**跨平台**的，能在Windows、Linux等多种操作系统上运行。



4.1 第一个 JSP 页面



1、JSP概述

◆ JSP 与 HTML 区别

- ✓ HTML 页面是静态页面，也就是事先由用户写好放在服务器上，由 Web 服务器向客户端发送。
- ✓ JSP 页面是由 JSP 容器执行该页面的 Java 代码部分，然后，实时生成的 HTML 页面，Java 代码部分不会在客户端显示出来。

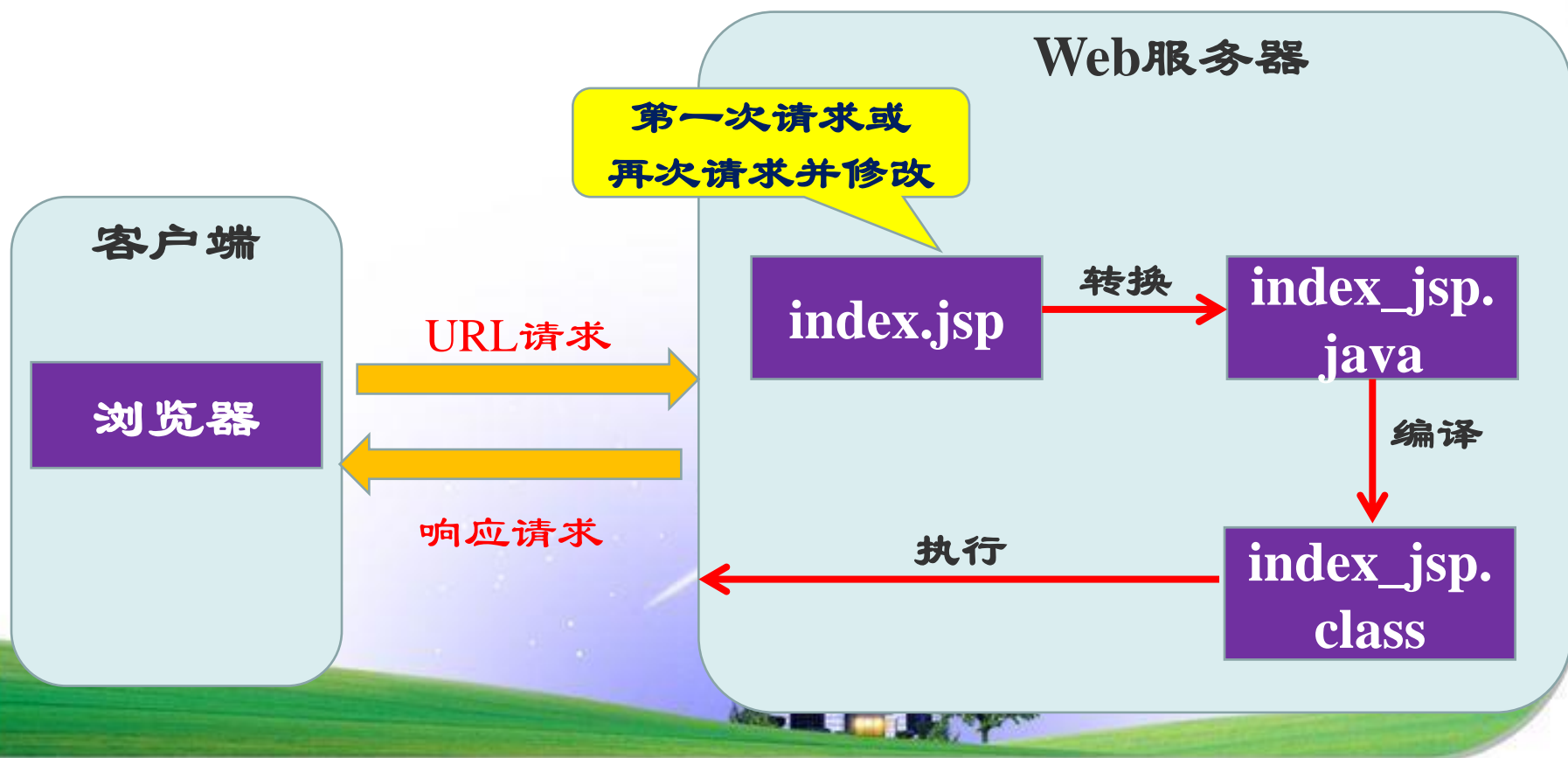


4.1 第一个 JSP 页面



2、JSP 工作流程

- ◆ 情况1：第一次请求一个页面
- ◆ 情况2：该页虽然是再次请求，但已经过修改

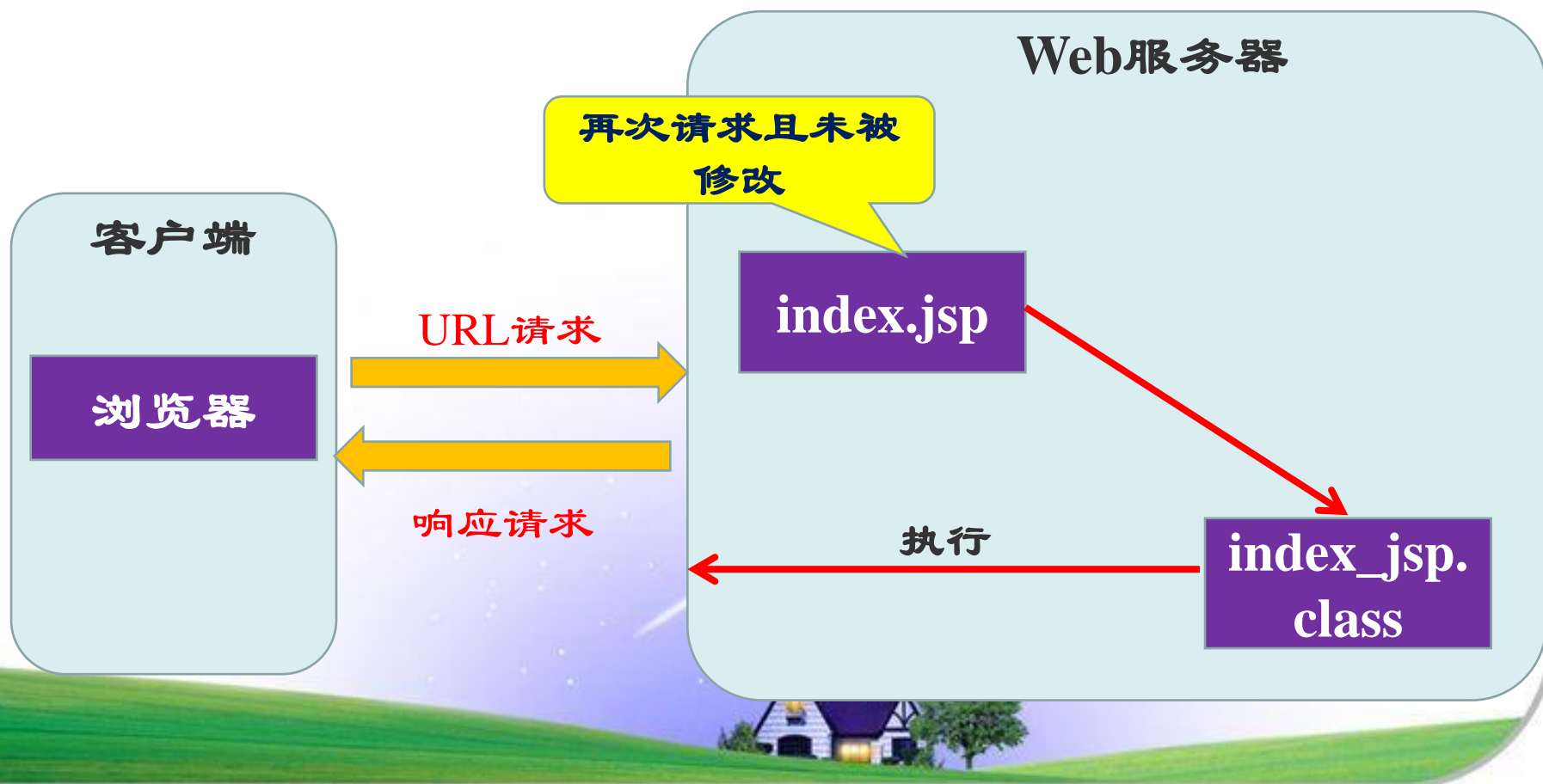


4.1 第一个 JSP 页面



2、JSP 工作流程

- ◆ 情况3：该页面是再次请求且未被修改



4.1 第一个 JSP 页面



3、JSP源代码的基本构成

- ◆ **静态部分**：指的是HTML、JavaScript、CSS等用来显示数据以及设置数据的显示样式的部分，这些代码JSP服务器不会处理，而是直接传送到客户端由浏览器自动解析。
- ◆ **动态部分**：指的是包括脚本、指令、动作、注释在内用来完成数据处理的部分，这些代码将由JSP服务器直接处理，然后将处理后的结果传送到客户端由浏览器再次解析。



4.1 第一个 JSP 页面



3、JSP源代码的基本构成

The diagram illustrates the structure of a JSP page, showing the source code and its rendered output.

Source Code (example.jsp):

```
1 <!-- JSP 示例 -->
2
3 <%@ page language="java" contentType="text/html; charset=gb2312"%>
4
5 <html>
6   <head><title>JSP页面的基本构成</title></head>
7   <body>
8
9     <jsp:include page="index.jsp"></jsp:include>
10
11     <%! int n=5; %>
12
13     <%
14       int mul=1;
15       for(int i=1;i<=n;i++)
16         mul=mul*i;
17       out.print(n+"!=");
18     %>
19
20     <%=mul%>
21
22   </body>
23 </html>
```

Dynamic Part (动态部分): Indicated by red boxes, it includes the page directive (`<%@ page ... %>`) and the dynamic content (`<jsp:include ... >` and the Java code block `<%! ... %>`).

Static Part (静态部分): Indicated by blue boxes, it includes the HTML structure (`<html>`, `<head>`, `<body>`) and the static content (`<%=mul%>`).

Rendered Output: The browser window shows the output of the JSP page, displaying the title "JSP页面的基本构成" and the calculated value "5!= 120".

4.1 第一个 JSP 页面



3、JSP源代码的基本构成

- ◆ JSP脚本：指的是在JSP文件中插入的**Java代码**，主要用来进行业务逻辑处理，通过**<% %>**来定义。JSP脚本在服务器端按从上到下的顺序执行。

```
<%! int n=5; %>
```

```
<%
```

```
    int mul=1;  
    for(int i=1;i<=n;i++)  
        mul=mul*i;  
    out.print(n+"!=");
```

```
%>
```

```
<%=mul%>
```


4.1 第一个 JSP 页面



3、JSP源代码的基本构成

- ◆ JSP指令：是为JSP服务器设计的，用来**设置**与整个**JSP页面相关属性**的命令。JSP指令在整个页面范围内有效，且不在客户端产生任何可见输出。

```
<%@ page language="java" contentType="text/html; charset=gb2312" %>
```



4.1 第一个 JSP 页面



3、JSP源代码的基本构成

- ◆ JSP动作：是为了简化JSP中的业务逻辑而简化的一组操作，实际上就是已**封装好的Java脚本**，使用它们可以动态的引用文件、使用JavaBean组件、传递参数、请求转发等。

```
<jsp:include page="index.jsp"></jsp:include>
```





课堂练习4.1

◆ 填空

✓ JSP源文件是在传统网页的HTML代码中插入____和____而构成的文件，其扩展名为____。

✓ JSP源代码的动态部分包括____、____、____和注释。

◆ 简答

✓ 请简述JSP的工作流程。





课堂练习4.1

◆ 选择

- ✓ 代码`<%@ page language="java" contentType="text/html; charset=gb2312" %>`是属于 ()
A.JSP注释 B.JSP脚本 C.JSP指令 D.JSP动作
- ✓ 代码`<jsp:include page="index.jsp"></jsp:include>`是属于 ()
A.JSP注释 B.JSP脚本 C.JSP指令 D.JSP动作
- ✓ 代码`<% int n=5; %>`是属于 ()
A.JSP注释 B.JSP脚本 C.JSP指令 D.JSP动作





4.2 注释

- ◆ **JSP的注释是在JSP源文件中出现的、用来解释说明JSP源代码的部分。**
- ◆ **注释的内容不参与编译和执行，不受JSP的语法限制，不会显示在客户端页面中。**
- ◆ **使用注释的目的：帮助读者快速了解这段程序的功能。**





4.2 注释

◆ 分类

- ✓ 能够发给客户端的注释，即会在客户端源代码文件中显示注释内容。
 - HTML 注释
- ✓ 不能发送给客户端的注释，即不会在客户端源代码文件中显示其内容，仅提供给程序员阅读。
 - JSP 注释：不会保留在Servlet中
 - Java 代码注释：会保留在Servlet中





4.2 注释——HTML注释

◆ 语法：<!-- HTML注释 -->

```
<%@ page language="java" contentType="text/html; charset=gb2312" %>
<html>
  <body>
    <%
      out.print("欢迎来本系统! ");
    %>
    <br>
    <!-- HTML风格注释, 它会发送到客户端-->
  </body>
</html>
```

HTML注释未显示
在客户端网页中

HTML注释显示
在了客户端源代码文件中

欢迎来本系统!

<html>

<body>

欢迎来本系统!

<!-- HTML风格注释, 它会发送到客户端-->

</body>

</html>



4.2 注释——HTML注释

- ◆ **注意：**在HTML注释中可加入JSP表达式，从而动态生成注释内容。

```
<%@ page language="java" contentType="text/html; charset=gb2312"
import="java.util.*" %>
```

```
<!-- 代码生成时间: <%=new Date()%>-->
```

```
<html>
```

```
<body>
```

```
<%
```

```
out.print("欢迎来本系统! ");
```

```
%>
```

```
<br>
```

```
</body>
```

```
</html>
```

每次执行后查看页面源代码时，该注释部分内容均不同

```
<!-- 代码生成时间: Sun Oct 22 01:56:51 CST 2017-->
<html>
  <body>
    欢迎来本系统!
    <br>
  </body>
</html>
```




4.2 注释——JSP注释

◆ 语法： `<%-- JSP注释 --%>`

```
<%@ page language="java" contentType="text/html; charset=gb2312" %>
<html>
  <body>
    <%
      out.print("欢迎来本系统! ");
    %>
    <br>
    <%-- JSP风格注释, 它不会发送到客户端 --%>
  </body>
</html>
```

JSP注释未显示在
客户端网页中

欢迎来本系统!

```
1
2 <html>
3   <body>
4     欢迎来本系统!
5     <br>
6
7   </body>
8 </html>
```

JSP注释未显示在客
户端源代码文件中



4.2 注释——Java代码注释

◆ 语法：① **//**单行注释 ② **/*** 多行注释 ***/**

```
<%@ page language="java" contentType="text/html; charset=gb2312" %>
<html>
  <body>
    <%
      out.print("欢迎来到本系统! "); //Java注释
    %>
    <br>
  </body>
</html>
```

Java代码注释未显示在客户端网页中

欢迎来本系统!

Java代码注释未显示在客户端源代码文件中

```
1
2 <html>
3   <body>
4     欢迎来到本系统
5     <br>
6   </body>
7 </html>
```



课堂练习4.2

◆ 填空

- ✓ JSP的注释包括三类：_____、_____和_____。
- ✓ HTML注释的符号是_____, JSP注释的符号是_____, Java脚本注释包括两类，其中符号//是_____, 符号/**/是_____。
- ✓ 若需将注释内容发送到客户端，则需使用_____来注释，若不需要将注释的内容发送到客户端，而仅仅是保留在Servlet文件中，则需使用_____来注释，若既不需要将注释的内容发送到客户端，也不需要保留在Servlet文件中，则需要使用_____来注释。



JSP脚本元素

- ◆ JSP脚本：在JSP文件中插入的**Java代码**，在**服务器端运行**，用**<% %>**定义。
- ◆ JSP脚本元素可以分为三类
 - ✓ **JSP脚本段**：用于处理JSP页面所涉及功能的业务逻辑；
 - ✓ **JSP表达式**：用于在JSP页面中输出表达式的值；
 - ✓ **JSP声明**：用于在JSP页面中定义变量和方法。



4.4 JSP程序段

- ◆ JSP 程序段就是插入到 JSP 程序的 Java 代码段，可以实现定义变量等一系列复杂的逻辑操作。
- ◆ 语法格式：<% 合法的java代码 %>

```
<%@ page language="java" contentType="text/html; charset=gb2312"%>
```

```
<html>
```

```
<body>
```

```
<%
```

```
    for (int i = 1; i <= 5; i++) {  
        out.println("欢迎来到本系统<br>");
```

```
    }
```

```
%>
```

```
</body>
```

```
</html>
```

欢迎来到本系统
欢迎来到本系统
欢迎来到本系统
欢迎来到本系统
欢迎来到本系统

```
<html>
```

```
<body>
```

```
    欢迎来到本系统<br>
```

```
    欢迎来到本系统<br>
```

```
    欢迎来到本系统<br>
```

```
    欢迎来到本系统<br>
```

```
    欢迎来到本系统<br>
```

```
</body>
```

```
</html>
```



4.4 JSP程序段

- ◆ 说明1：JSP中输入信息到客户端使用out.print()系列方法而非System.out.print()系列方法。

```
<%@ page language="java" contentType="text/html; charset=gb2312"%>
<html>
  <body>
    <%
      for (int i = 1; i <= 5; i++) {
        System.out.println("欢迎来到本系统<br>");
      }
    %>
  </body>
</html>
```





4.4 JSP程序段

- ◆ 说明2：不能在 JSP 程序段中定义方法。

```
<%@ page language="java" contentType="text/html; charset=gb2312"%>
<%
    void print(){
        for (int i = 1; i <= 5; i++) {
            out.println("欢迎来到本系统<br>");
        }
    }
%>
<html>
    <body>
        <% print(); %>
    </body>
</html>
```



4.4 JSP程序段

- ◆ 说明3： 在一个JSP页面中可以编写任意数量的脚本片段，脚本段会在服务器端按顺序依次自动执行。

```
<%@ page language="java" contentType="text/html; charset=gb2312"%>
```

```
<html>
```

```
<body>
```

```
<%
```

```
    for (int i = 1; i <= 5; i++) {
```

```
%>
```

```
        欢迎来到本系统 <br>
```

```
<%
```

```
    }
```

```
%>
```

```
</body>
```

```
</html>
```

欢迎来到本系统
欢迎来到本系统
欢迎来到本系统
欢迎来到本系统
欢迎来到本系统

```
<html>
```

```
<body>
```

```
欢迎来到本系统 <br>
```

```
欢迎来到本系统 <br>
```

```
欢迎来到本系统 <br>
```

```
欢迎来到本系统 <br>
```

```
欢迎来到本系统 <br>
```

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




课堂练习4.3

◆ 编程

- ✓ 编写一个显示“九九乘法口诀表”的JSP程序，并要求在程序中进行适当的注释。
- ✓ 编写一个计算并显示1至100累加和的JSP程序，并要求在程序中进行适当的注释。



4.3 JSP表达式

- ◆ 语法：<%=变量/返回值/表达式%>
- ◆ 作用：将其里面内容所运算的结果输出到客户端

```
<%@ page language="java" contentType="text/html; charset=gb2312"%>
<html>
```

```
<body>
```

```
<%
```

```
String name = "Jack";
```

```
String msg= "欢迎来到本系统! ";
```

```
%>
```

```
<br>
```

```
<%=name+", "+msg%>
```

```
</body>
```

```
</html>
```

Jack, 欢迎来到本系统!

```
<html>
```

```
<body>
```

```
<br>
```

```
Jack, 欢迎来到本系统!
```

```
</body>
```

```
</html>
```

```
<%out.print(name+", "+msg); %>
```

4.3 JSP表达式

- ◆ 说明：JSP程序段和JSP表达式可混合使用，使代码组成更为灵活。

```
<%@ page language="java" contentType="text/html; charset=gb2312"%>
<html>
  <body>
    <%
      for (int i = 1; i <= 10; i++) {
        <%=i%>:欢迎来到本系统<br>
      }
    %>
  </body>
</html>
```

1:欢迎来到本系统
2:欢迎来到本系统
3:欢迎来到本系统
4:欢迎来到本系统
5:欢迎来到本系统
6:欢迎来到本系统
7:欢迎来到本系统
8:欢迎来到本系统
9:欢迎来到本系统
10:欢迎来到本系统

4.3 JSP表达式

◆ 使用 JSP 表达式，需要注意几个细节：

✓ JSP 表达式中不能用 ";" 结束

`<%=i++; %>` ✗

✓ 在 JSP 表达式中不能出现多条语句

✓ JSP 表达式的内容一定是字符串类型，或者能通过 `toString()` 函数转换成字符串的形式





课堂练习4.4

◆ 填空

- ✓ 与JSP脚本段`<% out.print(n+1); %>`功能等价的JSP表达式为 `<%=n+1%>`。

◆ 编程

- ✓ 编写一个显示“九九乘法口诀表”的JSP程序，要求用JSP表达式完成页面输出。
- ✓ 编写一个计算并显示1至100累加和的JSP程序，要求用JSP表达式完成页面输出。





4.5 JSP声明

- ◆ JSP 声明用来定义程序中使用的实体，如变量、方法和类。
- ◆ 语法格式：**<%! 变量/方法/类的声明 %>**

◆ 举例：**<%! String str="欢迎"; %>**

```
<%!  
    public int add(int x,int y){  
        return x+y;  
    }  
%>
```

```
<%!  
    public class Student{  
        String name;  
        Student(String name){  
            this.name=name;  
        }  
    }  
%>
```





4.5 JSP声明

- ◆ 说明1: JSP 声明中定义的变量、方法和类是全局性的, 在 JSP 页面中的任何地方都能够使用。

```
<%@ page language="java"
contentType="text/html;
charset=gb2312"%>
<html>
  <body>
    <%
      out.println(str);
      String str = "欢迎";
    %>
  </body>
</html>
```



```
<%@ page language="java"
contentType="text/html;
charset=gb2312"%>
<html>
  <body>
    <%
      out.println(str);
    %>
    <%!
      String str = "欢迎";
    %>
  </body>
</html>
```





4.5 JSP声明

- ◆ 说明2：JSP 声明中不能使用out.print()系列方法做输出操作。

```
<%@ page language="java" contentType="text/html; charset=gb2312"%>
<html>
  <body>
    <%!
      out.println("欢迎来到本系统");
    %>
  </body>
</html>
```



4.5 JSP声明

- ◆ 说明3：JSP 声明中定义的变量在服务器关闭后释放，JSP脚本段中定义的变量在页面关闭后释放。

```
<%@ page language="java" contentType="text/html; charset=gb2312" %>
<html><head><title>全局变量和局部变量</title></head>
```

```
<body>
```

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

全局变量

```
<%
```

```
int m=0;
```

局部变量

```
n++;
```

```
m++;
```

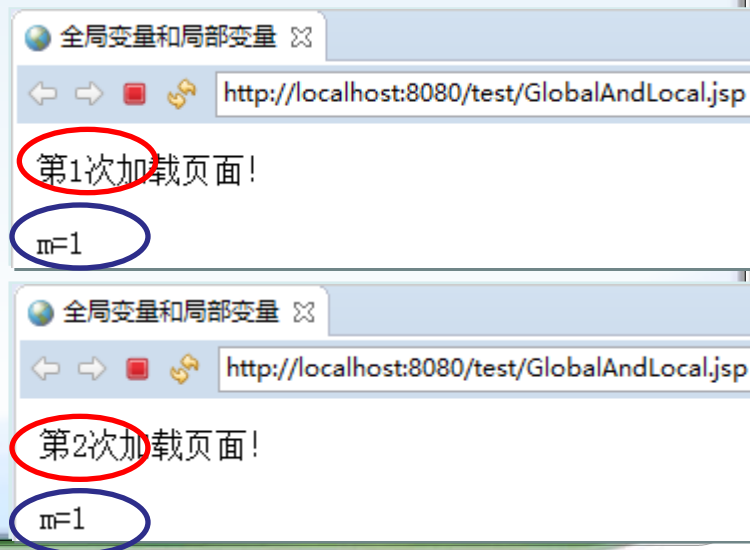
```
out.print("第"+n+"次加载页面！");
```

```
%>
```

```
<br><br>m=<%=m %>
```

```
</body>
```

```
</html>
```





课堂练习4.5

◆ 程序改错

```
<%@ page language="java" contentType="text/html; charset=gb2312"%>
<%
    void print(){
        for (int i = 1; i <= 5; i++) {
            out.println("欢迎来到本系统<br>");
        }
    }
%>
<html>
    <body>
        <% print(); %>
    </body>
</html>
```



课堂练习4.5

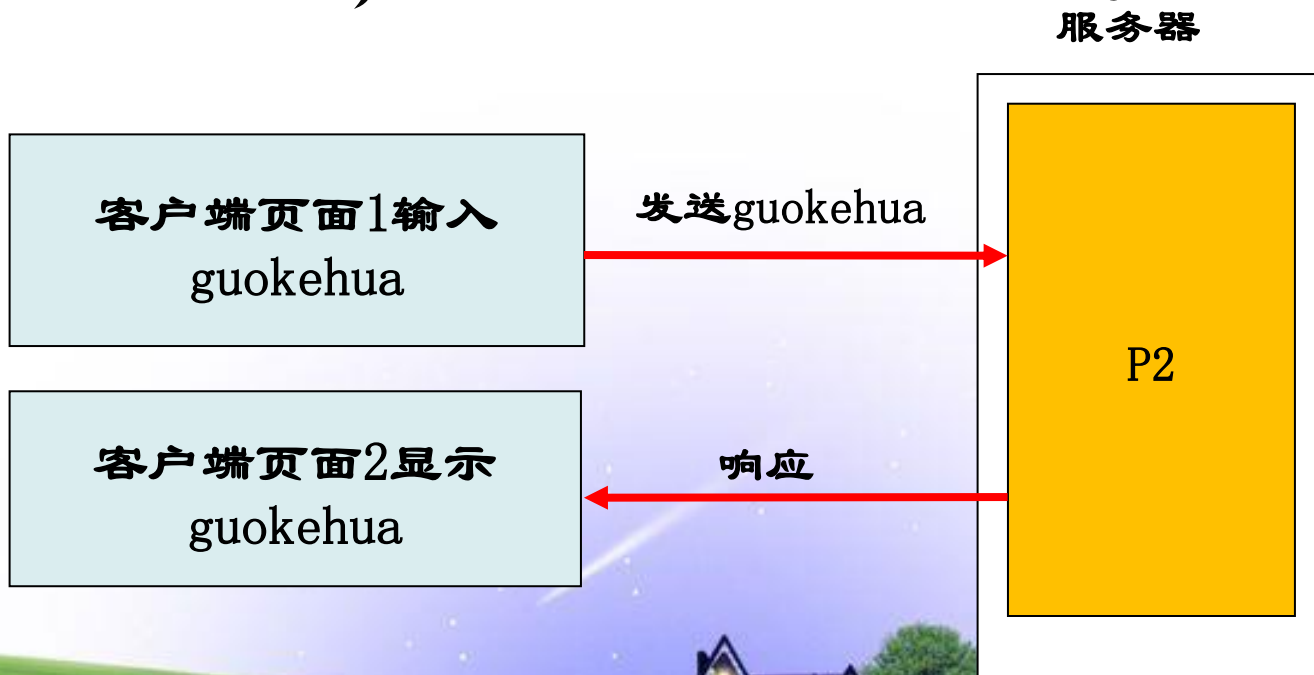
◆ 编程

- ✓ 声明一个全局的圆类，该类包含1个字段存放半径值，两个方法分别计算圆的周长和面积。
- ✓ 每次加载页面都自动创建一个圆对象（圆的半径自拟），在页面中输出圆的周长和面积。
- ✓ 在页面中输出自动生成的圆的个数。



4.6 URL传值

- ◆ HTTP 是无状态的协议，Web 页面本身无法向下一个页面传递信息，如果需要通过服务器中转。



4.6 URL传值

◆ 含参字符串格式：

URL?参数名1=参数值1&参数名2=参数值2.....

◆ 举例：

http://localhost:8080/proj04/page.jsp?m=3&n=5

注册





4.6 URL传值

◆ 获取传递到的参数的方法：

`request.getParameter("参数名")`

◆ 举例：

`<%`

`String m=request.getParameter("m");`

`String n=request.getParameter("n");`

`%>`

`page.jsp`

`<%`

`String name=request.getParameter("name");`

`String no=request.getParameter("no");`

`%>`

`handle.jsp`



```
<%@ page language="java" import="java.util.*" pageEncoding="gb2312"%>
<%
```

//定义一个变量:

```
String str = "12";
```

```
int number = Integer.parseInt(str);
```

```
%>
```

该数字的平方为: $\< \% = \text{number} * \text{number} \% \>$ <HR>

```
<a href="urlP2.jsp?number=< \% = \text{number} \% \>">到达p2</a>
```

urlP1.jsp

```
<%@ page language="java" import="java.util.*" pageEncoding="gb2312"%>
<%
```

//获得number

```
String str = request.getParameter("number");
```

```
int number = Integer.parseInt(str);
```

```
%>
```

该数字的立方为: $\< \% = \text{number} * \text{number} * \text{number} \% \>$ <HR>

urlP2.jsp

该数字的平方为: 144

[到达p2](#)

该数字的立方为: 1728





4.6 URL传值

◆ URL传值的局限性：

- ✓ 对传值的数据类型有限制：只能是字符串
- ✓ 保密性差：传输数据的值会在浏览器地址栏里面被看到，秘密性要求很严格的数据(如密码)，不应用URL方法来传值

◆ URL传值的优势：

- ✓ 简单性：用URL传值比较方便
- ✓ 平台支持的多样性：没有浏览器不支持URL

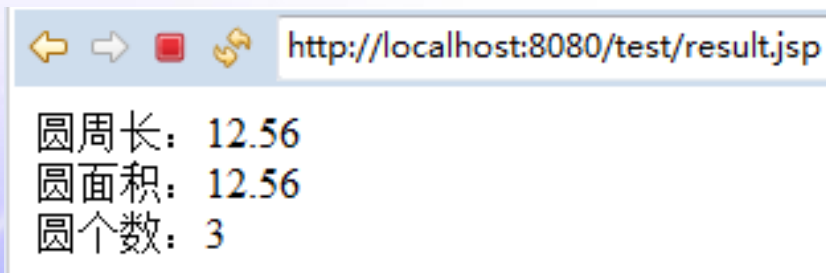
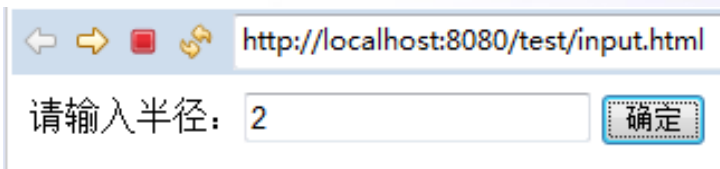




课堂练习4.6

◆ 在【课堂练习4.5】第二题的基础上编程实现如图所示页面，要求

- ✓ 编写查询页面input.html页，执行效果如左图所示，点击“确定”按钮跳转到结果页面result.jsp
- ✓ 编写页面result.jsp，执行效果如右图所示。





4.7.1 JSP指令

- ◆ JSP指令元素是指在JSP文件中包含在`<%@`与`%>`之间的部分，用来设置页面的相关信息，在服务器端执行，**不产生输出**，在整个**页面范围内有效**。
- ◆ 格式：`<%@ 指令 属性1=" 值"属性n=" 值" %>`
- ◆ 注意：
 - ✓ `<%@`与`%>`都是完整标记，中间不能添加空格
 - ✓ 指令名和属性名大小写是敏感的
 - ✓ 一条指令中可根据需要设置多个不同的属性，各属性间用空格分隔，属性值均为字符串类型





4.7.1 JSP指令

◆ JSP指令包括

- ✓ **page**指令：用于设定JSP页面的全局属性和相关功能
- ✓ **include**指令：用于将特定位置上的资源包含到当前的JSP文件中
- ✓ **taglib**指令：用于定义一个标签库及标签库的前缀



4.7.1 JSP指令



1、page指令

- ◆ 功能：用于定义整个JSP页面的相关属性，这些属性在JSP被服务器解析成Servlet时会转换为相应的Java程序代码。

- ◆ 语法格式：

`<%@ page 属性1=" 值"属性n=" 值" %>`

- ◆ 例如

```
<%@ page language="java" contentType="text/html; charset=gb2312" %>
```



4.7.1 JSP指令



1、page指令

- ◆ 说明1：一个JSP页面可只含一条page指令，在这条page指令中包含所有要设置的属性-值对，也可以将这些属性-值对分散在多条page指令中。

- ◆ 例如：

```
<%@ page language="java" contentType="text/html; charset=gb2312" %>
```



```
<%@ page language="java" %>  
<%@ page contentType="text/html; charset=gb2312" %>
```



4.7.1 JSP指令



1、page指令

- ◆ **说明2：** page指令可以放在JSP页面的任何位置，但为了便于阅读和格式规范，通常放到JSP页面的开头。
- ◆ **说明3：** 使用page指令时，可根据需要选择设置其中一个或几个属性的值，而非一定要设置所有属性



4.7.1 JSP指令



1、page指令

◆ 常用属性

属性	功能	属性	功能
language	指定JSP使用的脚本语言	info	定义JSP页面的描述信息
errorPage	指定当前页面运行异常时调用的页面	isErrorPage	说明当前页面是否为其他页面的异常处理页面。
import	导入使用的Java包	isELIgnored	指定是否忽略EL表达式
session	指定在当前页中是否允许session操作	buffer	指定处理页面输出内容时的缓冲区大小
contentType	设置返回浏览器网页的内容类型和字符编码格式	autoFlush	指定当缓冲区满时是否自动清空
pageEncoding	指定JSP页面的字符编码	isThreadSafe	指定是否线程安全



4.7.1 JSP指令



1、page指令——import属性

◆ 功能：用于设置导入本页Jsp脚本会用到的包和类。

◆ 语法格式：

✓ 导入指定类：<%@ page import="包.类" %>

✓ 导入指定包中所有类：<%@ page import="包.*" %>

◆ 例如：

```
<%@ page import="java.util.Date" %>
```

```
<%@ page import="java.util.*" %>
```



4.7.1 JSP指令



1、page指令——import属性

- ◆ 说明1：有4个包(javax.servlet.jsp.*、java.lang.*、javax.servlet.http.*、javax.servlet.*)是默认被载入JSP页面中的。若Jsp脚本用到的类不在这4个包中，则必须在import属性中设置它们的访问路径，否则会导致语法错误。

```
*PDImport.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8" %>
2
3 <html>
4 <head>
5 <title>显示服务器当前时间</title>
6 </head>
7 <body>
8 现在时间是: <%= new Date().toLocaleString() %>
9 </body>
10 </html>
```

语法错误，当前
页面找不到Date
类

4.7.1 JSP指令



1、page指令——import属性

- ◆ 说明1：有4个包(javax.servlet.jsp.*、java.lang.*、javax.servlet.http.*、javax.servlet.*)是默认被载入JSP页面中的。若Jsp脚本用到的类不在这4个包中，则必须在import属性中设置它们的访问路径，否则会导致语法错误。

```
*PDImport.jsp
1 <%@ page language="java" import="java.util.*" %>
2 contentType="text/html; charset=UTF-8" %>
3
4 <html>
5 <head>
6 <title>显示服务器当前时间</title>
7 </head>
8 <body>
9 现在时间是: <%=new Date().toLocaleString() %>
10 </body>
11 </html>
```

导入Date类所在
包中所有类

运行

显示服务器当前时间

http://localhost:8080/test/PDImport.jsp

现在时间是: 2017-4-30 13:57:39

4.7.1 JSP指令



1、page指令——import属性

- ◆ 说明2：import属性是page指令中**唯一可以多次出现的属性**，其他属性若多次出现则会导致编译错误。

```
<%@ page import="java.text.*, java.util.*" %>
```



```
<%@ page import="java.text.*" %>
```

```
<%@ page import="java.util.*" %>
```



4.7.1 JSP指令

1、page指令——pageEncoding属性

- ◆ 功能：用于定义JSP源文件中代码的编码格式，默认值为"*ISO-8859-1*"。支持中文的编码格式有"*gb2312*"、"*GBK*"、"*UTF-8*"等。
- ◆ 语法格式：`<%@ page pageEncoding = " 值" %>`
- ◆ 例如：

```
<%@ page pageEncoding = " GBK " %>
```



4.7.1 JSP指令



1、page指令——errorPage属性

- ◆ 功能：当JSP页面运行异常时，用于指定处理错误的另一个JSP页面，其值为URL字符串。
- ◆ 语法格式：<%@ page errorPage = " 值" %>
- ◆ 例如：

```
<%@ page errorPage = "anErrorPage.jsp" %>
```



4.7.1 JSP指令

1、page指令——isErrorPage属性

- ◆ 功能：设置当前JSP页面为另一个JSP页面的错误处理页面，其值为布尔类型，默认值为false。
- ◆ 语法格式：<%@ page isErrorPage = " 值" %>
- ◆ 例如：

```
<%@ page isErrorPage = "true" %>
```
- ◆ 注意1：该属性是否设置为true不影响当前错误处理页面的执行，但设置为true后，服务器会根据原页面错误类型将相应http状态码返回到客户端，否则将返回执行正常的状态码。



```
<%@ page contentType='text/html; charset=gb2312'
errorPage='pageTest2_error.jsp'%>
```

pageTest2.jsp

```
<html>
```

```
<body>
```

```
<%//此页面会向pageTest2_error抛出异常，让其来处理
```

```
int num1=10;
```

```
int num2=0;
```

```
int num3=num1/num2;
```

```
%>
```

```
</body>
```

```
</html>
```

← → http://localhost:8080/test/pageTest2.jsp

网页出现数学运算异常!

地址栏仍显示原
页面地址而非错
误处理页面地址

```
<%@ page contentType='text/html; charset=gb2312' isErrorPage='true'%>
```

```
<html>
```

```
<body>
```

```
<% //此页面会会处理pageTest2.jsp抛出的异常
```

```
//友好地显示错误信息
```

```
out.println("网页出现数学运算异常!");
```

```
%>
```

```
</body>
```

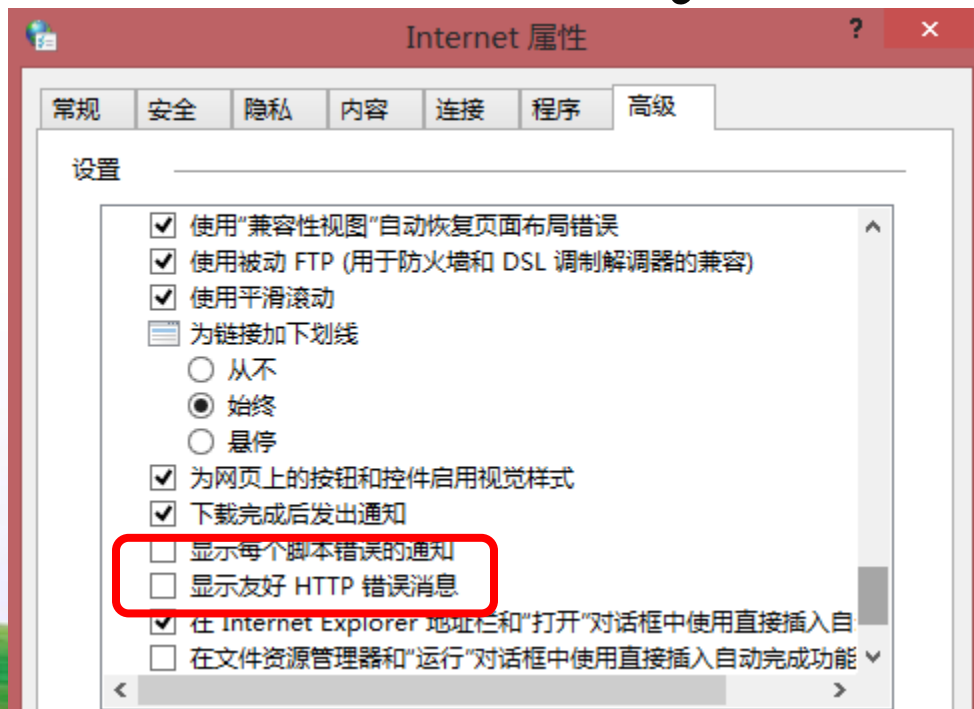
```
</html>
```

pageTest2_error.jsp

4.7.1 JSP指令

1、page指令——isErrorPage属性

- ◆ **注意2：**通常情况下，当页面执行异常会显示IE的默认错误界面而非自定义的错误处理页面，因此，需要修改“Internet选项→高级”中的“显示友好http错误消息”选项为不选中。



4.7.1 JSP指令



1、page指令——language属性

- ◆ 功能：用于定义JSP页面使用的语言，该属性的默认值是Java，目前也**只支持Java**。
- ◆ 语法格式：<%@ page language = " 值" %>
- ◆ 例如：

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



4.7.1 JSP指令



1、page指令——language属性

- ◆ 注意：若将language属性的值设置为其他语言，编译时会出现异常。

PDLanguage.jsp

```
1 <%@ page language="C#" %>
2
3 <html>
4 <head>
5 <title>page指令: la
6 </head>
7 <body>
8 <% out.print("langu
9 </body>
10 </html>
```

运行

http://localhost:8080/test/PDLanguage.jsp

HTTP Status 500 – Internal Server Error

Type Exception Report

Message /PDLanguage.jsp (line: 1, column: 2) Page directive: invalid language attribute

Description The server encountered an unexpected condition that prevented it from fulfilling the request.

Exception

4.7.1 JSP指令



1、page指令——language属性

- ◆ 注意：若将language属性的值设置为其他语言，编译时会出现异常。

```
PDLanguage.jsp
1 <%@ page language="java" %>
2
3 <html>
4 <head>
5 <title>page指令: language属性</title>
6 </head>
7 <body>
8 <% out.print("language属性的使用示例"); %>
9 </body>
10 </html>
```

运行

pageæ±žæ€šçš„ā½ç™çº°ā¾

http://localhost:8080/test/PDLanguage.jsp

languageæ±žæ€šçš„ā½ç™çº°ā¾

4.7.1 JSP指令

1、page指令——contentType属性

- ◆ 功能：用于指定JSP页面的MIME类型及字符编码，浏览器将根据此属性值解析发送到客户端的信息。

- ◆ 语法格式：

`<%@ page contentType="MIME 类型; charset=字符编码"%>`

- ✓ MIME类型：text/html(默认)、image/gif等

- ✓ 字符编码：ISO-8859-1(默认)、UTF-8等

- ◆ 例如：

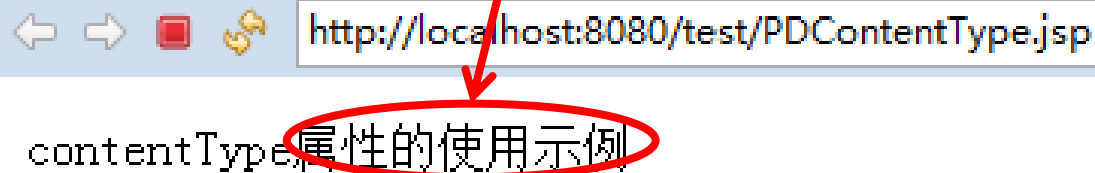
```
<%@ page contentType="text/html; charset=gb2312" %>
```



4.7.1 JSP指令

1、page指令——contentType属性

```
<%@ page contentType="text/html; charset=gb2312"%>
<html>
  <head>
    <title>page指令：contentType属性</title>
  </head>
  <body>
    <% out.print("contentType属性的使用示例"); %>
  </body>
</html>
```

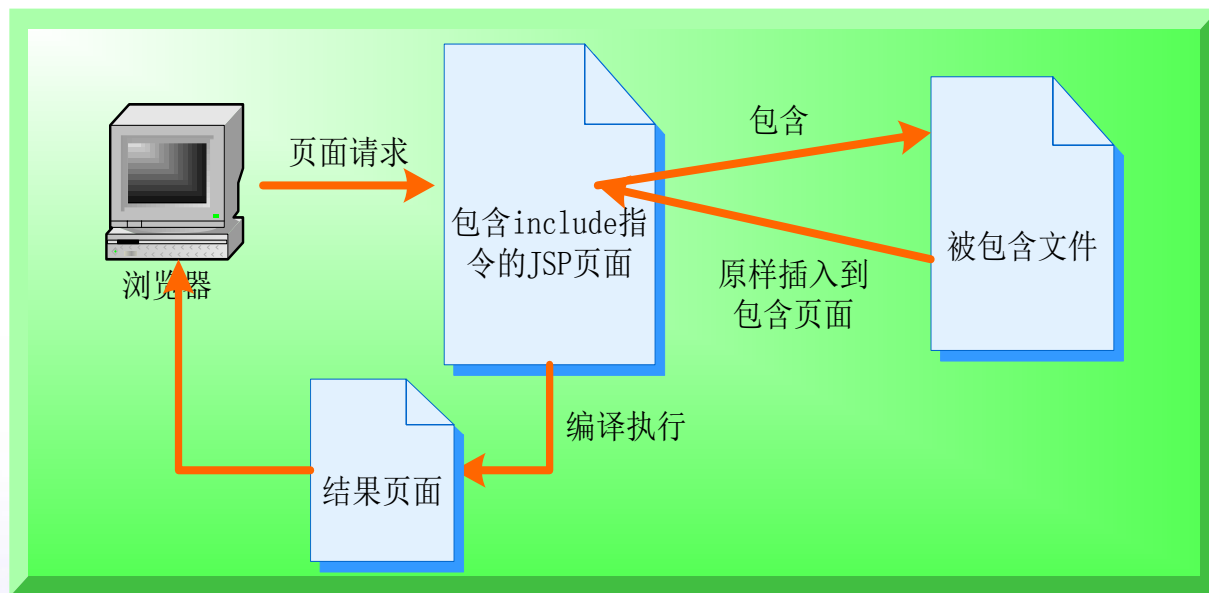


4.7.1 JSP指令



2、include指令

- ◆ 功能：向当前页面指定位置插入一个静态文件。



- ◆ 语法格式：

`<%@ include file=" 被包含文件的URL" %>`

- ◆ 例如：

```
<%@ include file=" logo.jsp" %>
```

```
<%@ page language="java" contentType="text/html; charset=gb2312"%>
<html>
  <body>
    <%
      out.print("欢迎来到本系统!");
    %>
    <br>
    <%@ include file="info.jsp" %>
  </body>
</html>
```

← → 🛑 🔄 http://localhost:8080/test/includeTest1.jsp ▾

includeTest1.jsp

欢迎来到本系统!

公司电话号码:010-89574895, 欢迎来电!

info.jsp页面运行
结果

```
<%@ page contentType="text/html; charset=gb2312"%>
<hr>
<center>
  公司电话号码:010-89574895, 欢迎来电!
</center>
```

info.jsp

4.7.1 JSP指令



2、include指令

- ◆ 说明：include指令在编译时对于文件包含到本页面，等价于代码复制。

includeTest1.jsp

```
1 <%@ page language="java" contentType="text/html; charset=gb2312"%>
2 <html>
3 <body>
4 <%@ include file="info.jsp"%>
5 <%
6 String msg="欢迎!";
7 %>
8 </body>
9 </html>
```

语法错误，与info.jsp
页面中的msg重名

info.jsp

```
1 <%@ page contentType="text/html; charset=gb2312"%>
2 <%
3 String msg="欢迎来到本系统!";
4 %>
```


4.7.1 JSP指令



2、include指令

- ◆ 说明：include指令在编译时对于文件包含到本页面，等价于代码复制。

```
includeTest1.jsp  info.jsp
1  <%@ page language="java" contentType="text/html; charset=gb2312"%>
2  <html>
3  <body>
4  <%@ page contentType="text/html; charset=gb2312"%>
5  <%
6      String msg="欢迎来到本系统!";
7  %>
8  <%
9      String msg="欢迎!";
10 %>
11 </body>
12 </html>
```

<%@ include file="info.jsp" %>
功能就是将info.jsp页面中的内容复制到include指令的位置



4.7.1 JSP指令



2、include指令

- ♦ 优点：include指令可以大大提高代码的重用性，且编译以后代码的维护和升级。

- ♦ 注意：

- ✓ 被包含的文件名不能是变量

```
<% String url="info.jsp" ;%>
```

```
<%@include file='<%=url%>' %>
```



- ✓ 不能向包含的文件传递参数

```
<%@include file="url.jsp? id=1" %>
```



4.7.1 JSP指令



3、taglib指令

- ◆ **功能：**在JSP页面中标识并引用指定标签库，同时指定标签的前缀(为方便使用给标签库起的别名)。这样，就可以通过前缀来引用标签库中的标签。

- ◆ **语法格式：**

`<%@ taglib uri="标签库的URI" prefix="标签前缀" %>`

- ◆ **例如**

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
```



4.7.1 JSP指令



3、taglib指令

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```
<html>
```

```
<head><title>taglib指令示例</title></head>
```

```
<body>
```

```
<c:out value="taglib 指令示例" />
```

```
</body>
```

```
</html>
```



http://localhost:8080/test/taglib.jsp

taglib 指令示例

```
<%= "taglib 指令示例" %>
```

或

```
<%out.print(" taglib 指令示例");%>
```



课堂练习4.7

◆ 填空

- ✓ 在JSP指令中，用于设定JSP页面的全局属性和相关功能的是____指令，用于将特定位置上的资源包含到当前的JSP文件中的是____指令，用于定义一个标签库及标签库的前缀的是____指令。
- ✓ 若要使客户端网页正常显示中文，则需要设置____指令的____属性，将属性中的字符编码设置为____。
- ✓ 我们需要设置____指令的____属性，将值设置为____，才能使用专门的错误处理页面error.jsp来处理JSP文件的异常。





课堂练习4.7

◆ 填空

- ✓ 使用的指令<% _____ %>可以在当前页面中导入类java.util.Date。
- ✓ 使用的指令<% _____ %>可以在当前页面中包含文件head.jsp。
- ✓ 使用的指令<% @ _____ %>可以在当前页面中导入SQL标签库(<http://java.sun.com/jsp/jstl/sql>), 并为其取别名sql。





课堂练习4.7

◆ 选择

✓ 在page指令中，可以重复多次出现的属性是（ ）

A.language B.contentType C.import D.errorPage

✓ 以下指令正确的是（ ）

A. `<%@ page import=" java.text.* , java.util.*" %>`

B. `<%@ page contentType="UTF-8" %>`

C. `<%@ page contentType="text/html" %>`

D. `<%@ page language="C#" %>`





课堂练习4.7

◆ 选择

✓ 以下指令正确的是 ()

A. `<%@include file="One.jsp,Two.jsp" %>`

B. `<%@include file="One.jsp" %>`

C. `<%@include file="<%=urlString%>" %>`

D. `<%@include file="check.jsp?name=Jim" %>`





课堂练习4.7

◆ 选择

✓ 以下说法正确的是 ()

- A. 一条page指令可以导入多个包或类
- B. 一条include指令可以插入多个文件资源
- C. 一条taglib指令可以导入多个标签库文件
- D. 在使用page指令时必须对其所有的属性值进行设置

