

jsp基础

一、JSP注释

定义显示注释和隐式注释：

```
<!--客户端查看源文件可以看得见-->
<%--JSP中的注释，客户端无法看见--%>
<%
    //Java提供的单行注释（客户端无法看见）
    /*
        多行注释（客户端无法看见）
    */
%>
```

二、Scriptlet(脚本小程序)

(1) <%%>

可以定义局部变量，编写语句等。

(2) <%!%>

定义全局变量、方法、类。

注意：尽量不要用此Scriptlet定义方法和类，开发者往往会采用Java Bean的形式

(3) <%= %>

输出一个变量或者一个常量，也称“表达式输出”

表达式输出相对out.println()输出更好，更加方便。

三、Scriptlet标签

程序中如果使用过多的<%%>会使得程序混乱，所以在新版本的jsp中，可以使用下面的标签进行代替

```
<jsp:scriptlet>
    java scriptlet代码
</jsp:scriptlet>
```

四、page指令

定义：定义一个jsp页面的相关属性，包括设置MIME类型、定义需要导入的包、错误页的指定等。

语法：

```
<%@ page 属性="内容"%>
```

常用的属性有：

autoFlush:可以设置成true或者false，如果设置成ture，当缓冲区满时，客户端的输出会被刷新，如果设置成false，当缓冲区满时，将出现异常。

buffer:

contentType:

errorPage:

extends:

import:

info:

isErrorPage:

isThreadSafe:

language:

pageEncoding:

session:

(1) 设置页面MIME

为JSP页面指定编码

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

page指令中指定了要使用的开发语言为java，然后通过contentType进行MIME类型设置，

按照HTML文本文件（ text/html ）进行显示，页面编码是（ GBK ）

MIME类型：可以从Tomcat安装目录/conf/web.xml中可以查询到所有已知的MIME类型

（ 如果想让页面按照word文档显示，可以将上面代码中"text/html"改为"application/msword" ）

(2) 设置文件源码

```
<%@ page language="java" contentType="text/html";pageEncoding="GBK"%>
```

在JSP中，如果pageEncoding存在，那么JSP的编码将由pageEncoding决定否则将由contentType中的charset属性决定；如果两者都不存在，则将使用ISO-8859-1的编码方式。

上面是说的第一阶段，第二阶段会使用utf-8编码，第三阶段就是TOMCAT生成的网页，此时使用的才是contentType。

一般情况下只用设置pageEncoding设置编码就可以了。

(3) 错误页的设置

满足条件：

①指定错误出现的跳转页：通过<%@ page errorPage=".jsp文件"%>（一旦出现错误就跳转到.jsp文件中）

②错误处理页必须有明确的标识：通过<%@ page isErrorPage="true"%>（表示此页面可以处理错误）

注意：此跳转属于服务器端跳转，地址栏的地址没有变化。

有时候可能出现无法跳转的错误页，这是因为可能错误页也出现错误了，这个时候我们可以这样编写：

在psge属性设置后面加一条：<% response.setStatus(200); %> （表示本页没有错误）

(4) 数据库连接操作

实例：

```
<%@ page contentType="text/html" pageEncoding="GBK"%>
```

```
<%@ page iimport="java.sql.*"%> //导入SQL包
```

```
<html>
```

```
<head> <title>标题</title> </head>
```

```
<body>
```

```
<%!
```

```
//定义数据库驱动程序
```

```
public static final String DBDRIVER="com.mysql.jdbc.Driver";
```

```
//数据库连接地址
```

```
public static final String DBURL="jdbc:mysql://localhost:3306/数据库名";
```

```
public static final String DBUSER="root"; //数据库连接用户名，一般情况下都会是root
```

```
public static final String DBPASS="数据库连接密码";
```

```
%>
```

```
<%
```

```
Connection conn=null;
```

```
PreparedStatement pstmt=null;
```

```
ResultSet rs=null;
```

```
%>
```

```
<%
```

```
try{ //数据库操作中会出现异常，所以 用try、、catch语句处理
```

```
Class.forName(DBDRIVER);
```

```
conn=DriverManager.getConnection(DBURL,DBUSER,DBPASS);
```

```
String sql="数据库操作语句"; //根据需要编写数据库操作语句
```

```
pstmt=conn.prepareStatement(sql);
```

```
rs=pstmt.executeQuery();
```

```
}
```

```
%>
```

之后再根据需从数据表从提取数据：例如从数据表中提取一个人的姓名：String name=rs.getString(1);

（一个简单的数据连接实例：<http://blog.csdn.net/kirinlau/article/details/53182789>）

五、包含指令

include指令是文件加载指令，用于在JSP中插入一个包含文件或代码的文件，这个包含的过程是静态包含。

它把文件插入后与原来的JSP文件合并成一个新的JSP页面（注意：如果被插入的内容发送改变，则包含这个页面的JSP文件就需要重新编译）。

使用include指令可以方便的把在多个页面中要重复显示的内容抽取出来，大大的减少代码的重复量，方便我们对重复内容的维护。

include指令的语法格式：

```
<%@ include file="被包含文件地址" %>
```

include指令只有一个file属性，该属性用于指定插入到JSP页面当前位置的文件资源。（注意：插入的路径一般使用相对路径）

被插入文件可以是一个JSP文件、HTML文件、文本文件或一段Java代码，单要保证被插入的文件是可访问的。

实例：

定义包含页：

```
<%@ page language="java" contentType="text/html" pageEncoding="GBK"%>
<html>
<head>
<title>jsp:include</title>
</head>
<body>
<h1>静态包含操作</h1>
<%@ include file="test.html"%>
<%@ include file="test.jsp"%>
<%@ include file="test.inc"%>
</body>
</html>
```

定义test.html页面

```
<html>
<head>
<metacharset="UTF-8">
<title>test.html</title>
</head>
<body>
this is test.html file!
</body>
</html>
```

定义test.jsp页面

```
<html>
<head>
<title>test.jsp</title>
</head>
<body>
<%= "this is a include test ! " %>
</body>
</html>
```

定义test.inc页面

This is test.inc file!

结果为：

this is test.html file!

this is a include test !

This is test.inc file!

注：在以上的静态包含中，实际上是先将所包含的3个文件的内容导入到包含页中，然后在一起进行编译，最后再将以份整齐的内容展现给用户，即：先包含，再集中进行编译处理。

(动态包含参考：<http://blog.csdn.net/chentiefeng521/article/details/51802062>)

六 跳转指令

将一个用户的请求 (request) 从一个页面传递到另一个页面，完成跳转操作。

实例：

跳转前 A.jsp

```
<%  
    String username="WJ";  
%>  
<jsp:forward page="B.jsp">  
    <jsp:param name="name" value="<%=username%>" /> 设置参数，参数名  
</jsp:forward>
```

跳转后 B.jsp

```
<h2>  
    <%=request.getPrameter("name")%>    //根据跳转前参数名，进行参数接收，getPrameter ( ) 方法  
</h2>
```

上述跳转也属于服务器端跳转。

七 JSP+JDBC 用户登录程序实现

基本思路：

- 创建数据表存储用户名及密码信息
- 用户登录表单html
- 登录检查页jsp
- 登录成功提示页jsp
- 登录失败提示页jsp

实例参考：<http://blog.csdn.net/pangqiandou/article/details/52713827>