《JavaEE 实验手册》

Java 教研室

版 本 1.3

文 档 提 供：Java 教 研 室　孙 丽 萍

修 改 记 录

修改时间	修改人	修改内容
20012.8.15	王伟	文档创建
2016.3.7	孙丽萍	修改

目录

第十三章 标准标签、简单标签

一、内容概述

本章的教学内容是介绍 JSP 中简单标签，定制标签的基本实现方法与知识。

通过本章实验应该达到的目的：掌握如何自定义简单标签、定制标签并使用自定义标签完成自己需要的功能。

二、实验一 helloworld

2.1 实验目的

开发一个标签实现类，实现 Tag 接口

实现功能：打印输出 HTML 文本的空标签

2.2 准备

保证 Eclipse 已经正确配置 Tomcat 服务器，可以进行 JavaWeb 项目编写、编译与调试。（如果以上设置不正确，请参考《Eclipse 部署配置 Tomcat 手册》进行正确设置）

2.3 实验步骤

步骤一：

在 Eclipse 中新建 Dynamic project “13_hellotag”，在“13_hellotag /src”目录下新建 com.RequiredTag.java，此类实现 Tag 接口，类中有两个私有变量 PageContext pageContext 和 Tag parentTag，doStartTag()方法输出一串 HTML 字符（任意合法的 HTML）返回 SKIP_BODY 属性常量，因为此标签是不需要处理标签内容体的空标签，doEndTag()方法返回 EVAL_PAGE 属性常量，因为我们要处理剩余 JSP 页面

步骤二：

在“13_hellotag /WebContent/WEB-INF/”目录下新建 RequiredTag.tld 文件,编写标签库描述符，关键代码如下

```
<taglib>
    <uri>http://localhost:8080/ 13_hellotag/required</uri>
    <tlib-version>1.0</tlib-version>
    <jsp-version>2.0</jsp-version>
    <tag>
        <name>required</name>
        <tag-class>com.RequiredTag</tag-class>
        <body-content>empty</body-content>
    <tag>
</taglib>
```

步骤三：

在web.xml文件中配置部署描述符，

```
<taglib>
    <taglib-uri>
        http://localhost:8080/13_hellotag /required
    </taglib-uri>
    <taglib-location>/Web-INF/tags/RequiredTag.tld</taglib-location>
</taglib>
```

步骤四：

在“13_hellotag/WebContent”目录下新建requiredTag.jsp使用<%@taglib %>指令引入标签requiredTag

```
<%@taglib prefix="mytag" uri="http://localhost:8008/13_hellotag/required" %>
```

图 2-2

步骤五：

运行项目“13_hellotag”，在浏览器中输入http://lcoalhost:8080/13_hellotag/requiredTag.jsp,回车，效果图如下：

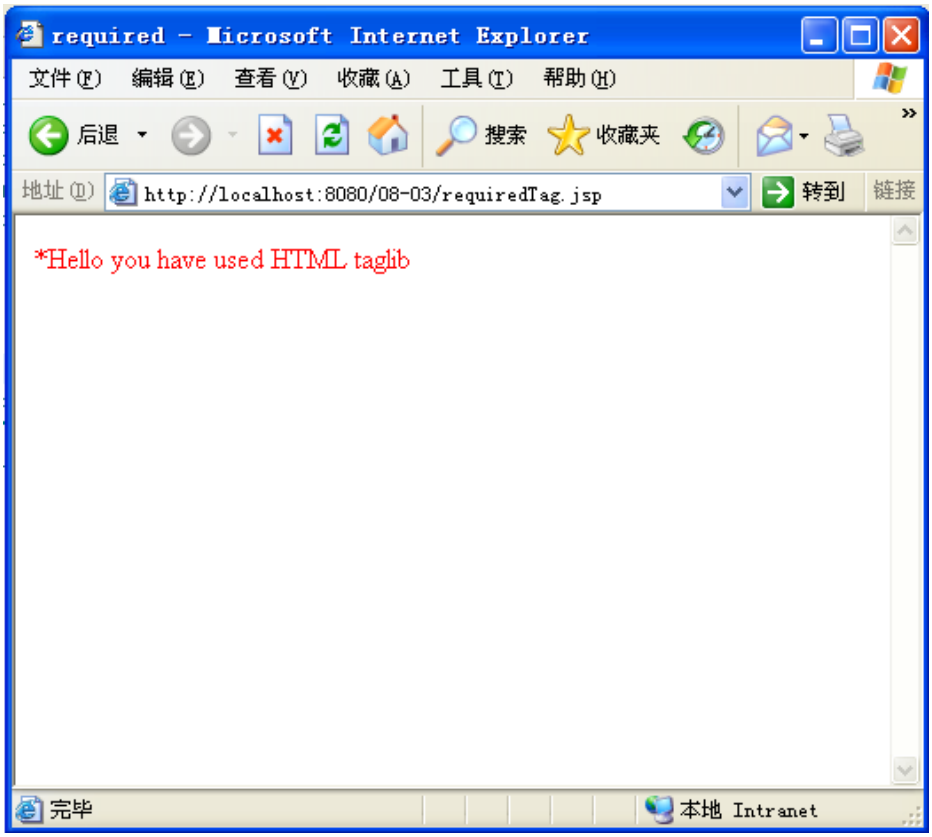

图 2-3

步骤六：

右键查看页面源代码，如下图

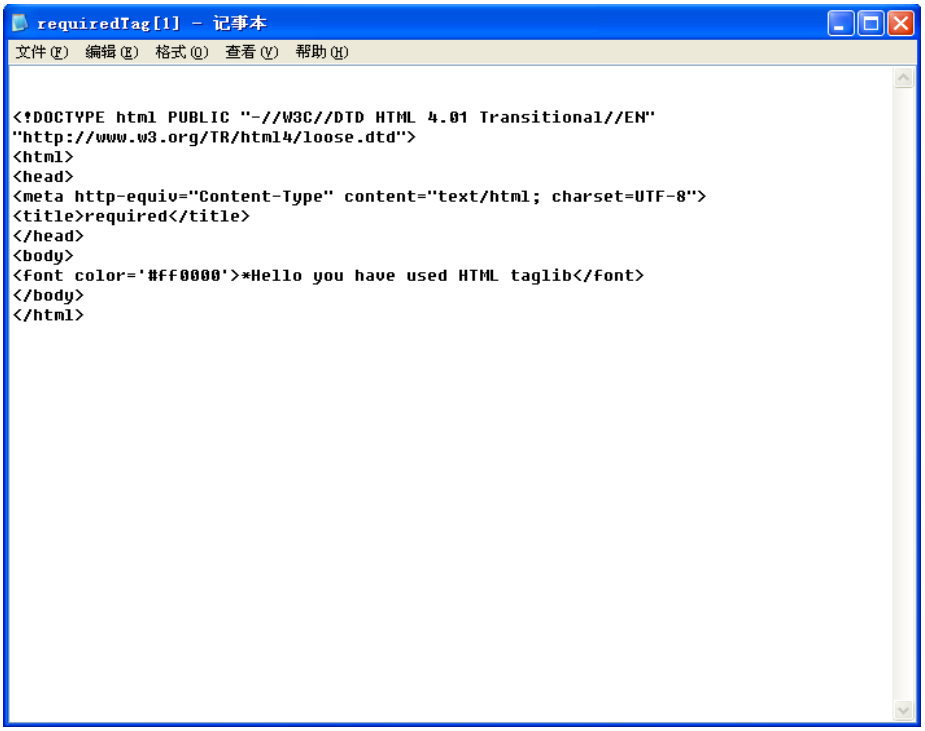

图 2-4

2.4实验结论

通过以上实验我们实现了最简单的自定义标签的功能，通过实验我们要掌握自定义标签的 4 个步骤编写标签实现类、编写标签库描述符 、编写部署描述符配置标签 、编写使用标签的 JSP 页面

三、实验二 自定义循环标签

3.1 实验目的

编写自定义标签，实现 IterationTag 接口
实现功能： 完成同 java 语言循环语句一样的功能，标签处理器 loop

3.2 准备

完成实验一

3.3 实验步骤

步骤一：

在“13_looptag/src”目录下新建 com.LoopTag.java,此类实现 IterationTag 接口，同样类有两个私有属性 pageContext 和 parentTag，还要定义一个私有变量 count，用于接收循环次数的标签属性值，每次调用 doAfterBody()方法时 count 值减一，然后返回 EVAL_BODY_AGAIN,直到 count 变量值为 0，如果 count 变量值为 0 返回 SKIP_BODY 来指明结束循环，关键代码如下

```
@Override
public int doAfterBody() throws JspException {
    // TODO Auto-generated method stub
    if (--count > 0)
        return EVAL_BODY_AGAIN;
    else
        return SKIP_BODY;
}
```

图 3-1

步骤二：

在“13_looptag/WebContent/WEB-INF/tags”目录下新建 loop.tld 标签库描述符文件,如下图

```
<taglib>
    <uri>http://localhost:8080/13_looptag/loop</uri>
    <tlib-version>1.0</tlib-version>
    <jsp-version>2.0</jsp-version>
    <tag>
        <name>loop</name>
        <tag-class>com.LoopTag</tag-class>
        <body-content>JSP</body-content>
        <attribute>
            <name>count</name>
            <required>true</required>
            <rtextprvalue>true</rtextprvalue>
        </attribute>
    </tag>
</taglib>
```

图 3-2

步骤三：

在 web.xml 文件中配置标签描述符文件

```
<taglib>
    <taglib-uri>http://localhost:8080/13_looptag/loop</taglib-uri>
    <taglib-location>/WEB-INF/tags/loop.tld</taglib-location>
</taglib>
```

图 3-3

步骤四：

在“13_looptag/WebContent”目录下新建 loop.jsp 用 taglib 伪指令引入 Loop 标签，并使用 loop 标签进行循环输出 helloworld，循环次数自己设定

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="loop" uri="http://localhost:8080/13_looptag/loop" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>loop</title>
</head>
<body>
    <loop:loop count="5">
        Hello World!<br>
    </loop:loop>
</body>
</html>
```

图 3-4

步骤五：

运行“13_looptag”项目,效果如下图

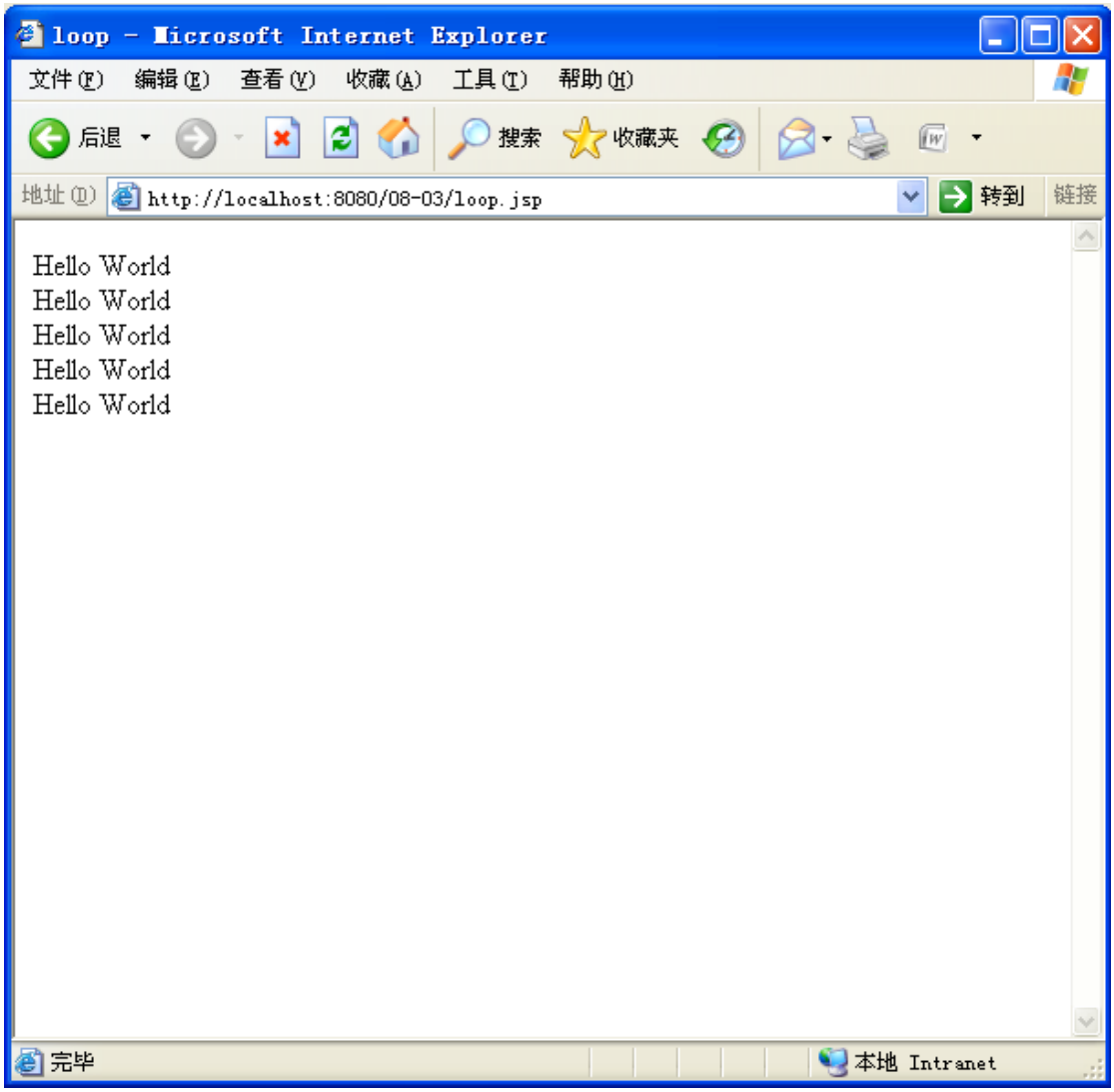

图 3-5

3.4 实验结论

通过以上实验我们再次练习自定义标签的编写步骤，熟悉标准标签 IterationTag 接口的使用

四、实验三 对字符串中 s 加粗显示

4.1 实验目的

编写自定义标签，实现 BodyTag 接口
实现功能：标签实现对指定字符进行加粗处理功能

4.2 准备

完成实验二

4.3 实验步骤

步骤一：

在“13_bodytag/src”目录下新建 MarkTag.java 文件，实现 BodyTag 接口，MarkTag.java 类有 PageContext pageContext、Tag parentTag、BodyContent bodyContent、String search，其中 search 表示要加粗的字符，bodyContent 表示所有输出到页面的内容。doStartTag()方法返回值为 EVAL_BODY_BUFFEREDS 属性常量，如果 doStartTag()方法返回值为 EVAL_BODY_INCLUDE 属性常量则 setBodyContent()方法就没有被调用所以会导致 doAfterBody()方法中抛出 NullPointerException 异常，为了实现对指定字符加粗，要对页面输出所有字符串进行循环判断，关键实现代码如下

```
public int doAfterBody() throws JspException {
    // TODO Auto-generated method stub
    try {
        JspWriter out = bodyContent.getEnclosingWriter();
        String text = bodyContent.getString();
        int len = search.length();
        int oldIndex = 0, newIndex = 0;
        while ((newIndex = text.indexOf(search, oldIndex)) >= 0) {
            if (newIndex < oldIndex) {
                break;
            }
            out.print(text.substring(oldIndex, newIndex));
            out.print("<b>" + search + "</b>");
            oldIndex = newIndex + len;
        }
        out.print(text.substring(oldIndex));
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return SKIP_PAGE;
}
```

图 4-1

步骤二：

在“13_bodytag/WebContent/WEB-INF/tags”目录下新建 mark.tld,编写标签库描述符，如图

```
<taglib>
    <uri>http://localhost:8080/13_bodytag/mark</uri>
    <tlib-version>1.0</tlib-version>
    <jsp-version>2.0</jsp-version>
    <tag>
        <name>mark</name>
        <tag-class>com.MarkTag</tag-class>
        <body-content>JSP</body-content>
        <attribute>
            <name>search</name>
        </attribute>
    </tag>
</taglib>
```

图 4-2

步骤三：

在 web.xml 文件中配置标签描述符文件，如下

```
<taglib>
    <taglib-uri>http://localhost:8080/13_bodytag/mark</taglib-uri>
    <taglib-location>/WEB-INF/tags/mark.tld</taglib-location>
</taglib>
```

图 4-3

步骤四：

在“13_bodytag/WebContent”目录下新建 mark.jsp,taglib 伪指令引入 mark 标签，并使用 mark 进行加粗标记，如下

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="mark" uri="http://localhost:8080/13_bodytag/mark" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loo
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>mark</title>
</head>
<body>
    <mark:mark search="s">
        she sells sea shells on the sea shore!
    </mark:mark>
</body>
</html>
```

图 4-4

步骤五：

运行“13_bodytag”项目,在浏览器中输入 http://localhost:8080/13_bodytag /mark.jsp,回车，运行效果如下图

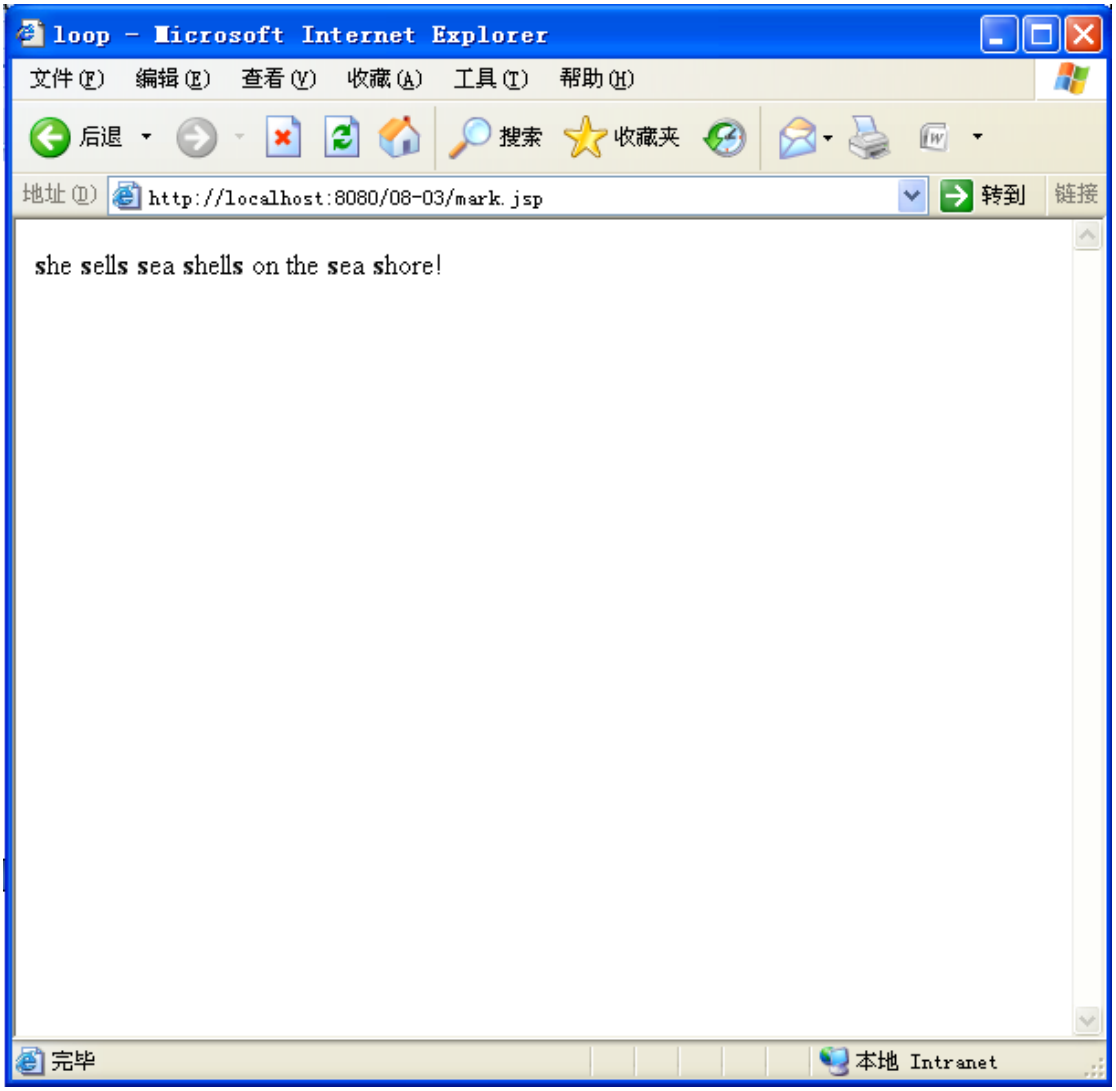

图 4-5

4.4 实验结论

通过以上实验复习了自定义标签编写步骤，熟悉了 BodyTag 接口使用，体会自定义标签的便捷之处