



Java程序设计

第05-01讲 自定义标签

武永亮

wuyongliang@edu2act.org



知识回顾/本讲先行知识



- JSTL标准标签库
 - JSTL核心标签库
 - `<c:out>`
 - `<c:if>`
 - `<c:forEach>`
 - JSTL格式化标签库
 - JSTL函数标签库
- JSTL标签库的使用
 - `jstl.jar`
 - `standard.jar`





本讲教学目标



- 掌握定制标签概念
- 掌握创建、使用简单标签
- 掌握创建、使用标准标签





本讲内容

- 定制标签简介
- 简单标签简介及使用
- 标准标签简介及使用
- JSP中使用标签文件



JSP代码片段



```
<body>
    <%@page import="com.DbConnection"%>
    <%
        DbConnection db = new DbConnection();
        String userName = request.getParameter("userName");
        String password = request.getParameter("password");
        boolean result = db.insert(userName, password);
        request.setAttribute("result", result);
    %>
    <c:if test="${result==true}">
        <c:redirect url="login.jsp" />
    </c:if>
    <c:if test="${result==false}">
        <c:redirect url="register.jsp" />
    </c:if>
</body>
```



为什么要定制标签



- JSP表示层中需要编写大量的JSP脚本代码
- 在其他JSP中要重复编写相同的代码，不能做到代码的复用
- JSP开发者可以创建新的标签来根据实际需求定制对应的方法——定制标签
- 使用自定义标签可以实现页面表示层的代码重用



几种常见的标签



- 空标签体的定制标签
 - `<prefix:tagName> </prefix:tagName>`
 - `<prefix:tagName />`
- 例如
 - `<mytag:required> </mytag:required>`
 - `<mytag:required />`



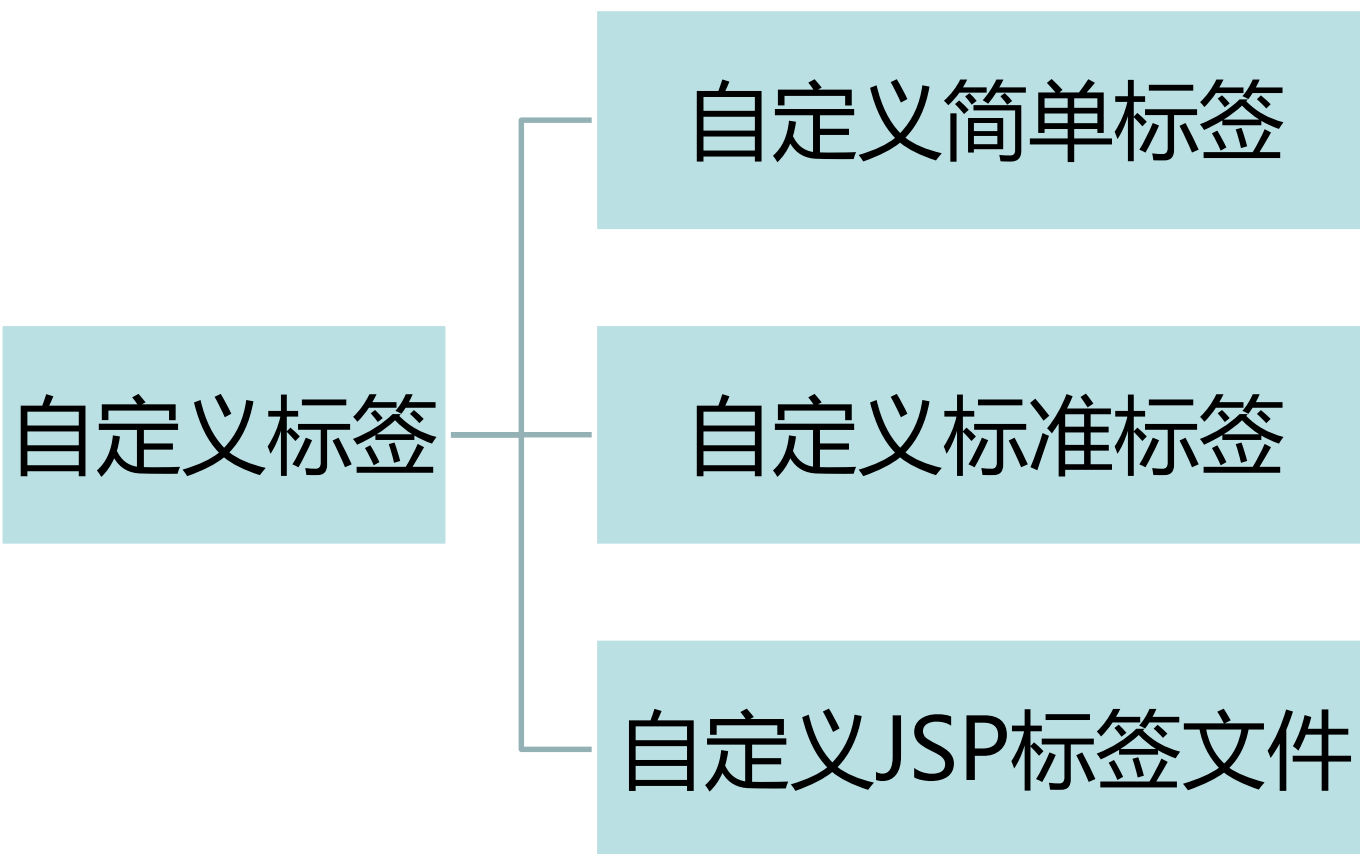
几种常见的标签



- 带属性的定制标签
 - 定制标签可以包含属性
 - `<前缀:标签名 属性1= "值1" 属性2= "值2" .../>`
- 例如:
 - `<test:function user="john" num="100"/>`



自定义标签的方法





本讲内容

- 定制标签简介
- 简单标签简介及使用
- 标准标签简介及使用
- JSP中使用标签文件



简单标签的概念



- JSP2.0提供了简单标签来快速实现自定义标签功能
- 简单标签提供了SimpleTag接口和SimpleTagSupport实现类



自定义简单标签的步骤



- 编写简单标签实现类
- 编写TLD文件
- 在web.xml中配置TLD文件
- 在JSP页面中使用taglib



1.编写简单标签实现类



```
public class SimpleTagExample
    extends SimpleTagSupport{
    public void doTag() throws
        JspException,IOException
    {
        getJspContext().getOut().print( "simple tag!" );
    }
}
```



2.编写标签库描述符文件

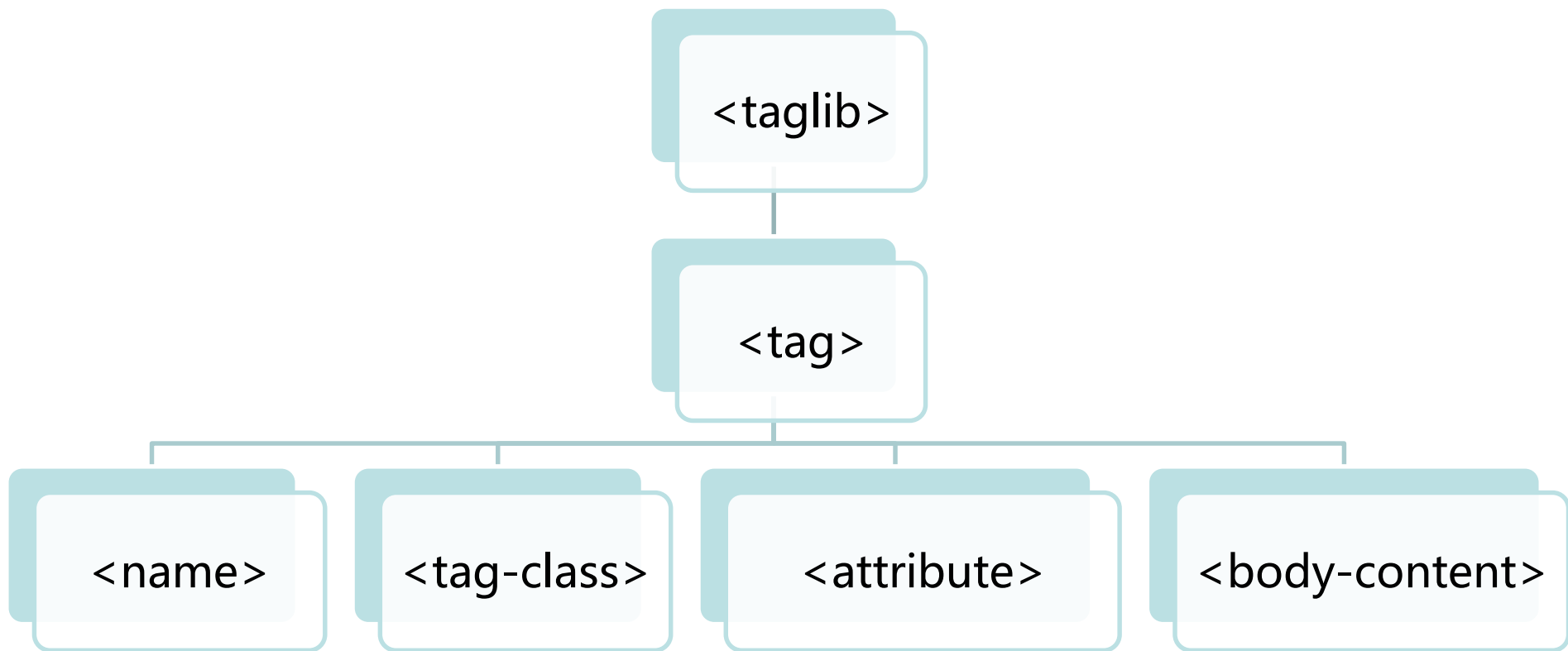


- 标签库描述符文件包含的信息可以指导JSP容器解析JSP的定制标签
- 一个标签库描述符就是一个XML文档，用于通知标签库使用者关于标签库的功能和用法

```
<?xml version="1.0" encoding="utf-8" ?>
<taglib>
...
<tag>
  <name>... </name>
  <tag-class>... </tag-class>
  <body-content>... </body-content>
  <attribute>    ...    </attribute>
</tag>
</taglib>
```



标签库描述符文件结构



标签库描述符文件结构



```
<taglib>
  <uri>http://localhost:8080/08-03/math</uri>
  <tlib-version>1.0</tlib-version>
  <jsp-version>2.0</jsp-version>
  <tag>
    <name>sqrt</name>
    <tag-class>com.MathTag01</tag-class>
    <body-content>empty</body-content>
    <description>
      Send a math expression to the JSP
    </description>
  </tag>
</taglib>
```



3.在web.xml中配置TLD文件



- `<jsp-config>` JSP相关配置
- `<taglib>` 定义在JSP中可使用的库
- `<taglib-uri>` 定义TLD文件的URI, JSP页面的`taglib`命令可以经由此URI获取到TLD文件
- `<taglib-location>` TLD文件所在的位置



3.在web.xml中配置TLD文件



```
<jsp-config>
  <taglib>
    <taglib-uri>http://localhost:8080/08-03/math</taglib-uri>
    <taglib-location>/MathTag01.tld</taglib-location>
  </taglib>
</jsp-config>
```



4.在JSP页面中使用taglib



```
<%@taglib prefix="math" uri=".../math"%>
```

```
...
```

```
    <math:sqrt />
```

```
...
```



简单标签实现方法



```
public class SimpleTagExample
    extends SimpleTagSupport{
    public void doTag() throws
        JspException,IOException
    {
        getJspContext().getOut().print( "simple tag!" );
    }
}
```



简单标签的接口



- SimpleTag接口

方法	功能描述
<code>setJspContext()</code>	使得JspContext对标签处理有效
<code>setParent()</code>	由Web容器调用使得父标签有效
<code>setJspBody()</code>	使得标签内容体对标签处理有效
<code>doTag()</code>	由容器调用开始简单标签的处理
<code>getParent()</code>	获取父标签

- SimpleTag生命周期

- 初始化, `setJspContext()`
- 处理标签内容体, `setBody()`
- 调用`doTag()`方法



简单标签类



- SimpleTagSupport类

方法	功能描述
<code>getJspContext()</code>	获取JspContext对象
<code>getJspBody()</code>	获取JspFragment对象
<code>findAncestorWithClass()</code>	获取指定类的祖先标签



简单标签的使用



- 空标签体的简单标签
 - 用于发送静态信息
 - MathTag01例子
- 带属性的简单标签
 - MathTag02例子
- 带标签体的简单标签
 - MathTag03例子





本讲内容

- 定制标签简介
- 简单标签简介及使用
- 标准标签简介及使用
- JSP中使用标签文件



标准标签



- 在JSP中可以通过简单标签来实现基本的标签自定义，能够完成简单的标签处理
- 对于复杂的用户需求标签，JSP提供了标准标签来进行实现
- 标准标签和简单标签的实现步骤基本一致，只是Java类继承的类不同。具体步骤如下：



标准标签的实现步骤



- 编写标准标签实现类
- 编写TLD文件
- 在web.xml中配置TLD文件
- 在JSP页面中使用taglib





1.编写标准标签实现类

- 编写标准标签实现类的两种方式：
 - 实现标准接口
 - 继承标准标签类

```
public class RequiredTag implements Tag {  
    @Override  
    public int doEndTag() throws JspException {}  
    @Override  
    public int doStartTag() throws JspException {}  
    @Override  
    public Tag getParent() {}  
    @Override  
    public void setPageContext(PageContext pageContext) {}  
    @Override  
    public void setParent(Tag parentTag) {}  
}
```



标准标签的实现步骤(同简单标签)



2.编写TLD文件

3.在web.xml中配置TLD文件

4.在JSP页面中使用taglib



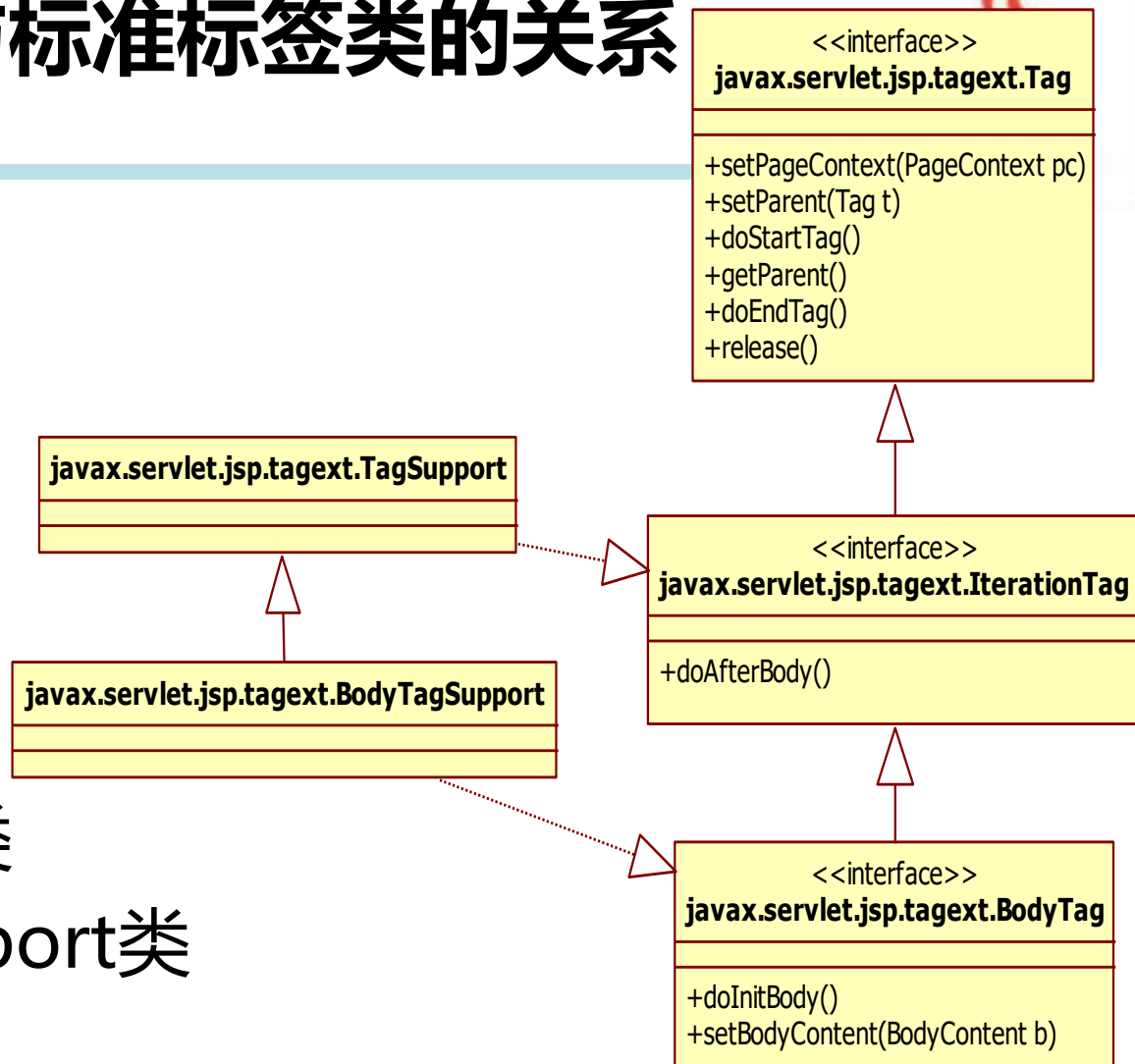
标准标签接口与标准标签类的关系

- 标准标签接口

- Tag
- IterationTag
- BodyTag

- 标准标签类

- TagSupport类
- BodyTagSupport类



Tag接口



- Tag接口方法

方法	功能描述
<code>int doStartTag()</code>	当标签被处理时调用
<code>int doEndTag()</code>	当标签结束处理时调用
<code>Tag getParent()</code>	获取此标签的直接父标签类
<code>void release()</code>	释放标签资源
<code>void setPageContext(pageContext)</code>	设置当前上下文
<code>void setParent(Tag)</code>	设置直接父标签类



Tag接口



- Tag接口常量属性

属性	功能描述
EVAL_BODY_INCLUDE	指明标签内容体需要处理
SKIP_BODY	指明标签内容体不需要处理
EVAL_PAGE	指明继续处理页面剩余部分
SKIP_PAGE	指明不继续处理页面剩余部分



IterationTag接口



- IterationTag接口方法
 - 继承自Tag接口具有Tag接口所有的方法
 - doAfterBody(), 执行完标签体内容调用此方法
- IterationTag接口属性
 - EVAL_BODY_AGAIN 指明标签内容体循环处理



BodyTag接口



- BodyTag接口方法
 - 继承自IterationTag接口
 - setBodyContent(), 将标签内容体实例传入标签
 - doInitBody()允许标签处理器初始化标签内容体



TagSupport与BodyTagSupport类



- TagSupport类
 - 实现了IterationTag接口，提供Tag和IterationTag接口所有方法的默认实现
- BodyTagSupport类
 - 继承自TagSupport类、实现了BodyTag接口
 - 继承了TagSupport类所有的方法并实现了BodyTag接口，提供setBodyContent()和doInitBody()方法的默认实现。新增getBodyContent()和getPreviousOut()处理缓冲区输出





本讲内容

- 定制标签简介
- 标准标签简介及使用
- 简单标签简介及使用
- JSP中使用标签文件



标签文件的概念



- 除简单标签和标准标签外，JSP2.0规范还提供了一种新的构建标签的形式：标签文件
- JSTL和表达式语言用于减少JSP中Java代码的数量，简单标签用于减少标签处理器实现类中Java代码的数量，使用标签文件可以不依赖Java代码，只要了解JSP语法就可以构建JSP页面中使用的定制标签



标签文件



- 标签文件是由JSP代码组成的，以.tag或.tags为后缀
- 标签文件可以包含表达式、JSP伪指令、标准标签和自定义标签
- JSP元素不能在标签文件中使用



使用标签文件



- 在JSP中使用 `<%@taglib prefix= “前缀” tagdir= “标签文件地址” %>` 来导入标签文件所在文件夹
- 通过前缀加标签文件名来使用一个标签文件，此处文件名不含文件后缀
- 例子：tag01.jsp
 - `<%@taglib prefix="test" tagdir="/WEB-INF/tags"%>`



标签文件的伪指令



- variable伪指令
 - 声明JSP变量<%@variable name-given= "x" %>
- tag伪指令
 - 对整个文件进行设置，类似page指令
- attribute伪指令
 - 指定属性为静态属性
 - fragment属性为true允许静态属性中插入JSP代码



标签文件的动作指令



- 标签文件包含了JSP中一系列的动作指令
- **jsp:invoke指令**
 - 使attribute伪指令中的fragment生效
- **jsp:doBody指令**
 - 用于处理标签内容体



小结



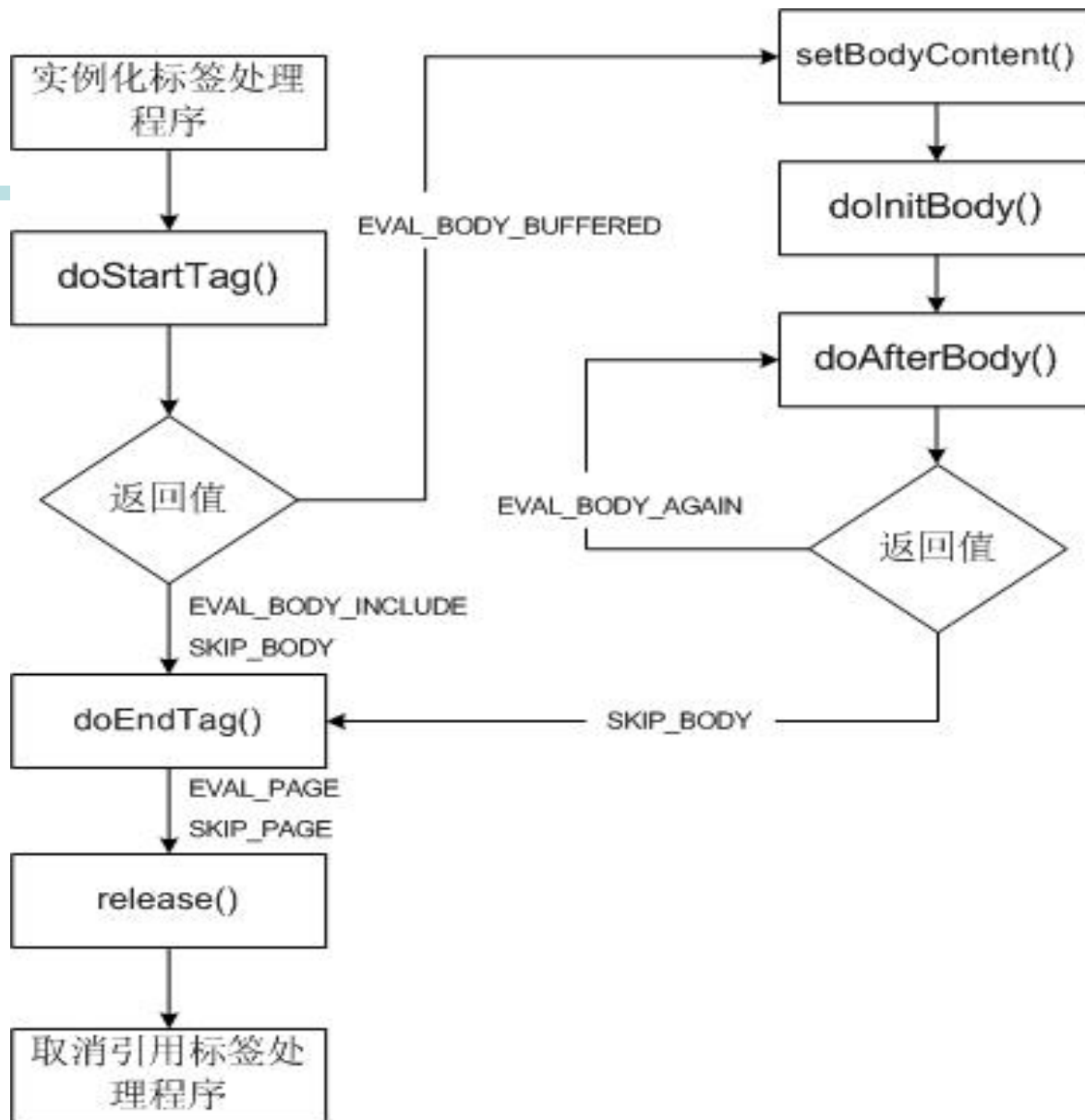
- 标签的主要应用场景
- 标准标签与简单标签编写、使用的步骤
- 标签文件的应用





实验

• 实验



本讲结束

- 谢谢大家

