



Java程序设计

第03-01讲 JSP简介

Java课程组



本讲教学目标



- 掌握JSP概念
- 熟悉手工在Tomcat中编写、部署、运行一个JSP程序
- 掌握Eclipse开发JSP程序
- 掌握JSP页面元素





知识回顾/本讲先行知识



- 掌握如何配置过滤器
- 掌握如何使用过滤器





本讲内容

- JSP简介
- 第一个JSP程序
- JSP页面元素
 - JSP的6种标签



JSP简介



- Java动态网页技术标准(Java Server Pages)
 - 是基于Servlet技术以及整个Java体系的Web开发技术
 - 是用于动态生成HTML文档的Web页面模板
 - JSP是为了改进Servlet输出动态的HTML复杂、烦琐等缺点而提出的,JSP允许Java代码与HTML写在同一页面中,使开发人员能快速的开发包括大量HTML代码的动态网页



动态网页技术



• 动态网页技术

ASP

ASP.net

PHP

JSP

有人做过试验，对ASP,PHP,JSP三种语言分别做循环性能测试及存取Oracle数据库测试。

在循环性能测试中，JSP只用了令人吃惊的四秒钟就结束了20000*20000的循环。而ASP、PHP测试的是2000*2000循环（少一个数量级），却分别用了63秒和84秒。（参考PHPLIB）。

性能

数据库测试中，三者分别对 Oracle 8 进行 1000 次 Insert,Update,Select,和Delete: Jsp 需要 13 秒，Php 需要 69 秒，ASP则 需要 73 秒。性能方面JSP的优势也是十分明显。

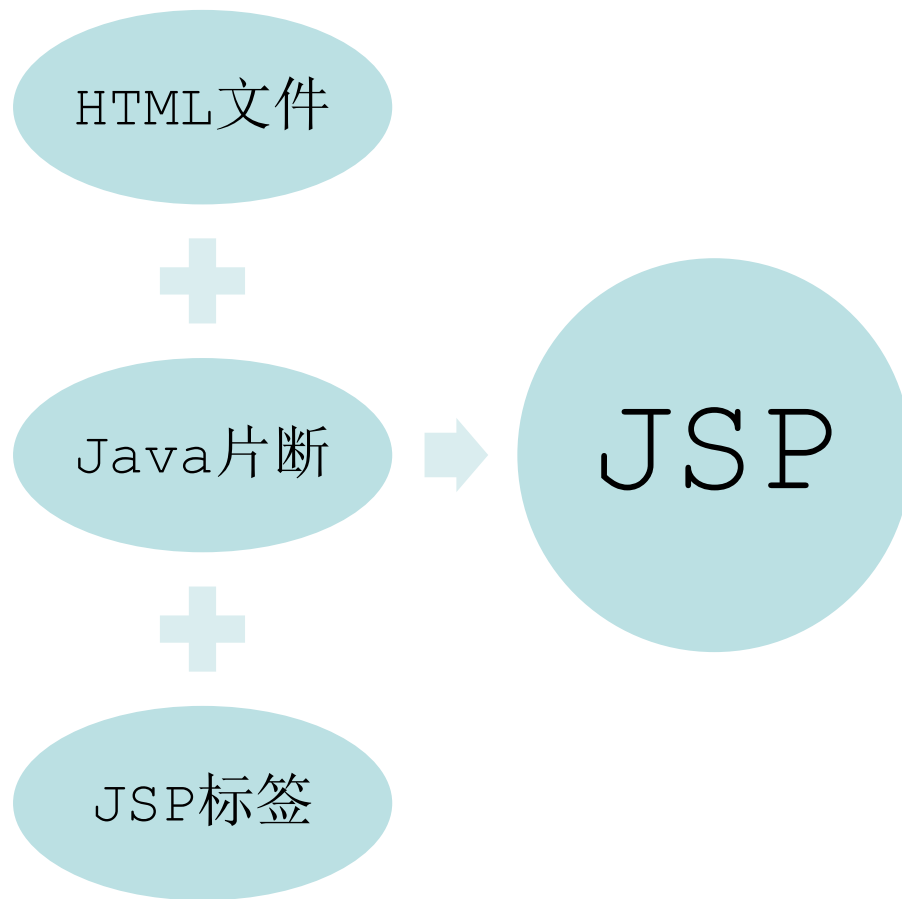
目前国内PHP与ASP应用最为广泛。而ASP.NET JSP由于是一种较新的技术，国内采用的较少。ASP.NET也在慢慢增多。但在国外，JSP已经是比较流行的一种技术，尤其是电子商务类的网站，多采用JSP。

四者中，JSP、PHP、ASP.NET都有着非常好的应用前景。JSP因为它的安全性，运行效率也有着很大的成长空间。

前景



JSP组成



JSP文件结构及主要标签



```
<%@ page contentType="text/html;charset=UTF-8"
    pageEncoding="UTF-8" %>
<%@ page import="java.util.* " %>
```

...

```
<html>
```

```
  <body>
```

其他 HTML 语言

```
<%
```

符合Java语法的Java语句

```
%>
```

其他 HTML 语言

```
</body>
```

```
</html>
```

JSP文件必须包括

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

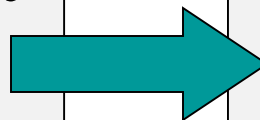


JSP示例(Hello.jsp)

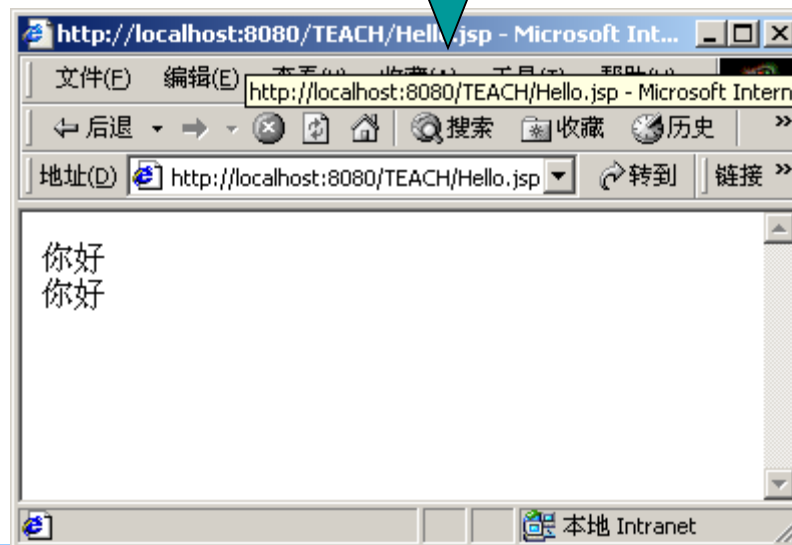


```
<%@ page contentType="text/html;  
charset=UTF-8" pageEncoding="UTF-8"%>
```

```
<html>  
<body>  
<%  
for ( int i=0; i<2; i++ )  
{  
>%  
    你好<br>  
<%  
}  
>%  
</body>  
</html>
```



```
<html>  
<body>  
    你好<br>  
    你好<br>  
</body>  
</html>
```





本讲内容

- JSP简介
- 第一个JSP程序
- JSP页面元素
 - JSP的6种标签



第一个JSP程序



- 实现功能：
 - 客户端输入名字
 - 返回客户端: "Hello: " + 名字

HTML

- 不能实现

Servlet

- `PrintWriter pw = response.getWriter();`
- `pw.write(""); ;`

JSP

- `${param.userName}`



JSP项目结构



- JSP必须依附项目（站点）
 - 一个动态站点包括WEB-INF目录（必须大写）
 - WEB-INF目录中包括web.xml文件
 - JSP文件与WEB-INF同级目录



Tomcat中部署JSP站点



- 站点直接放到Webapps目录下
 - Tomcat的Webapps目录是Tomcat默认的应用目录，当服务器启动时，会加载所有这个目录下的应用



Tomcat中部署JSP站点



- 在server.xml中
 - Host标签内创建Context

```
<Context path= "/myapp"
reloadable="true" docBase="D:\myapp"
workDir="D:\myapp\work"/>
```

 - 其中path是JSP应用虚拟路径
 - docBase是JSP应用的物理路径
 - workDir是JSP应用的工作目录(存放运行时生成的关于应用相关的文件)
 - 注：删除一个Web应用同时删除server.xml中相应的Context节点



Tomcat中部署JSP站点



- 创建一个Context文件
 - 直接在Tomcat的conf\Catalina\localhost目录下新建xml文件
 - 文件内容: `<Context path= "/myapp" reloadable="true" docBase="D:\myapp" workDir="D:\myapp\work"/>`
 - 注: 删除一个Web应用同时删除conf\catalina\localhost目录下相应的xml文件



Eclipse中编写、运行JSP项目



- 打开Eclipse, 新建Dynamic project “03-01”
- 在WebContent目录下新建index.jsp文件
- 编写index.jsp文件
- 运行 “03-01” 项目





本讲内容

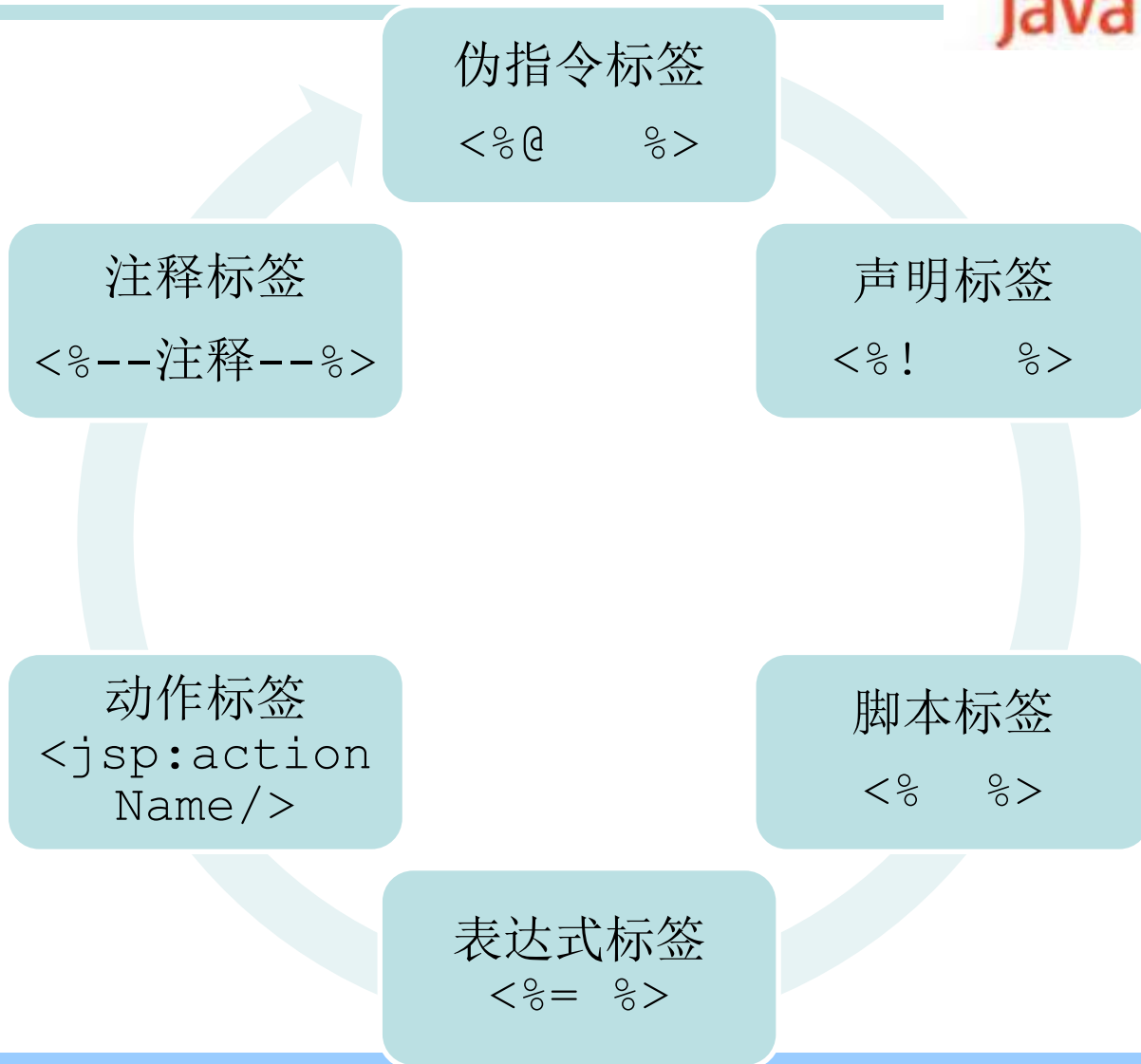
- JSP简介
- 第一个JSP程序
- JSP页面元素
 - JSP的6种标签



JSP页面元素



- HTML标签
- Java代码
- JSP标签



JSP伪指令



- 伪指令用来设置全局变量，声明类、要实现的方法以及输出内容的类型等

page

指定整个JSP页面的全局属性

```
<%@ page  
[attribute="value"  
.....] %>
```

include

通知jsp页面，内嵌指定页面,利用include可以将复杂的页面拆分成简单的页面

例:<%@ include
file="/header.html"%>

taglib

允许JSP页面开发者自定义标签

例:<%@ taglib
uri="taglibURI"prefix="perfix"%>



JSP声明指令



- 声明指令包括的是一段java代码
 - 例: `<%! int count =0;%>`
 - 例: `<%! String color[] =
{ "red" ," green" ," yellow" };
String getColor(int i){
 return color[i];
}
%>`



JSP脚本指令



- 脚本指令是镶嵌在JSP页面中的java代码
 - 主要用于处理客户端的请求
 - 例：

```
<% String name =  
request.getParameter("name");  
if (name != null) {  
    out.println("hello " + name);  
}  
%>
```



JSP表达式指令



- 表达式指令是JSP提供了一种简单方法用于访问可用的Java取值或其他表达式，还可用于给动作指令动态传递参数

例:<%=“hello world”%>

例:<% String name =“header.jsp”%>

<jsp: include page=“<%=name%>”%>

*注:表达式不能使用分号结尾

<%=name; %>



JSP注释指令



- 注释指令是增强代码的可读性

一种是只在JSP中显示

- `<%-- 注释内容--%>`

第二种是JSP脚本元素中的注释

- `//注释内容` 或者 `/* 注释内容*/`

第三种是JSP生成的HTML在客户端显示的注释

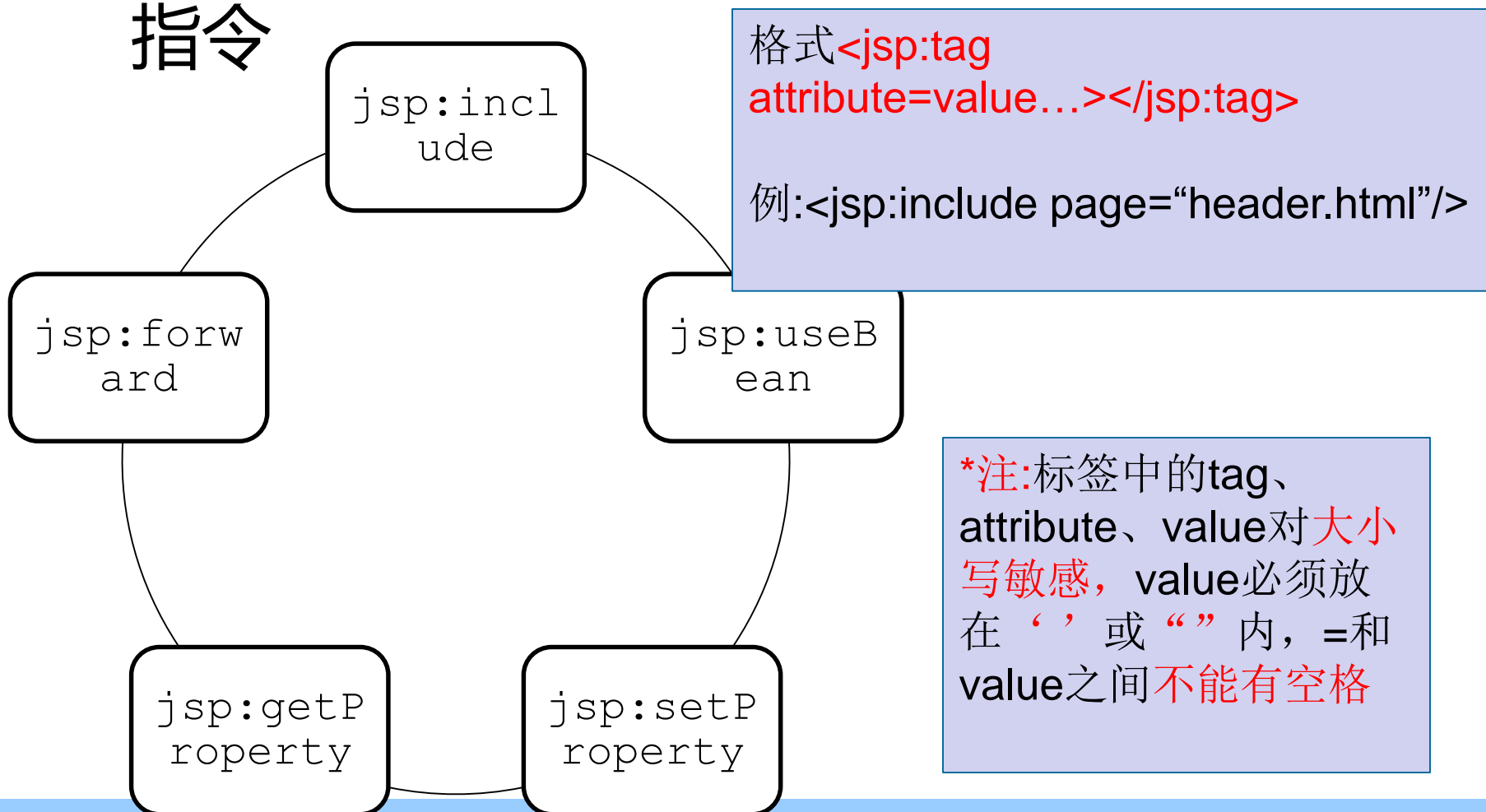
- `<!-- 注释内容 -->`



JSP动作指令



- 动作指令是在请求处理阶段向JSP容器发送指令



JSP标签注意事项



JSP页面中的伪指令元素会对整个页面产生作用

在合成的JSP页面中不能存在两个及以上的同名变量



小结



- JSP概念
- 手工在Tomcat中编写、部署、运行一个JSP程序
- Eclipse开发JSP程序
- JSP页面元素





实验



- 手工在Tomcat中编写、部署、运行JSP
- 在Eclipse里创建、编写、运行JSP
- 参考《03-01_实验手册.doc》手册



本讲结束

- 谢谢大家

