JSP:

1.

Java Server Pages

Java动态网页技术标准，为了改进Servlet输出动态的HTML复杂、繁琐等缺点提出的

2.JSP文件必须包括

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

3.三种部署方式

站点直接放到Webapps目录下

在server.xml中 Host标签内创建<Context path=”/myapp” reloadable=”true” docBase=”” workDir=””>

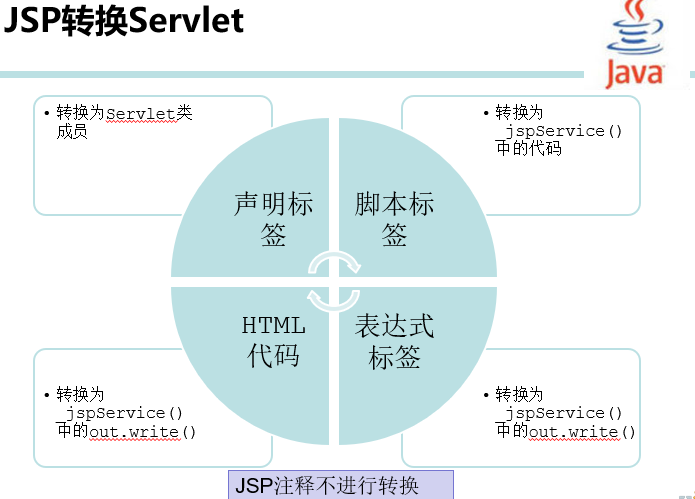
直接在Tomcat的conf\Catalina\localhost目录下新建xml文件<Context path=”/myapp” reloadable=”true” docBase=”” workDir=””>

4.JSP六种标签

* 伪指令标签 <%@ %>
* 声明标签<%! %>
* 脚本标签<% %>
* 表达式标签 <%= %>
* 动作标签<jsp:actionName/>
* 注释标签<%--注释--%>

5.JSP生命周期

转译（解析，校验，转换） 编译 加载 实例化 初始化 服务 销毁



JSP内置对象

Out application session request response

Page pageContext config exception

作用域范围

Page request session application

静态包含

在JSP源文件转换成对应Servlet源文件时将当前jsp文件和其包含的所有其他文件合并成一个Servlet文件

<%@include file=“地址”%>

动态包含

* include允许当前页面将客户端请求暂时转交给所包含的对象，一旦对象执行完毕，返回当前JSP页面

<jsp:include page=“地址” flush=“true”>

<jsp:param name=“” value=“”/>

动态包含中的包含文件和被包含文件各自独立编译、执行，彼此之间不能访问各自定义的变量(即page作用域范围内的对象)，只能共享request范围、session范围、application范围对象



* forward动作指令在调用转发操作后不会继续处理指令后的任何代码
* include动作指令在调用转发操作后，完成转发操作后将继续处理指令后的代码

EL表达式

El隐含对象

applicationScope sessionScope requestScope pageScope

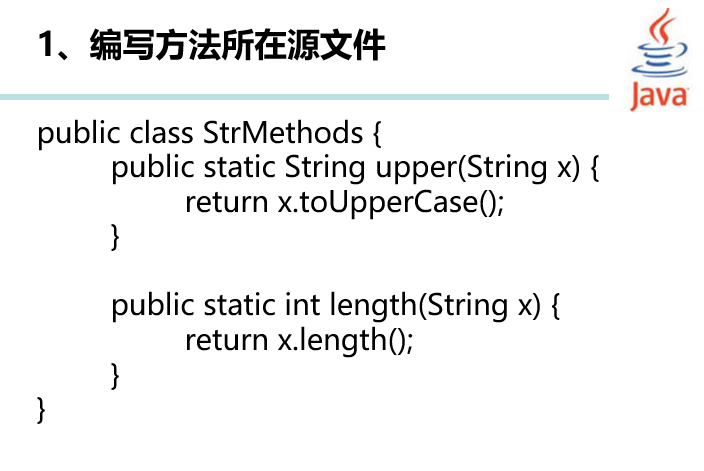
param paramValues

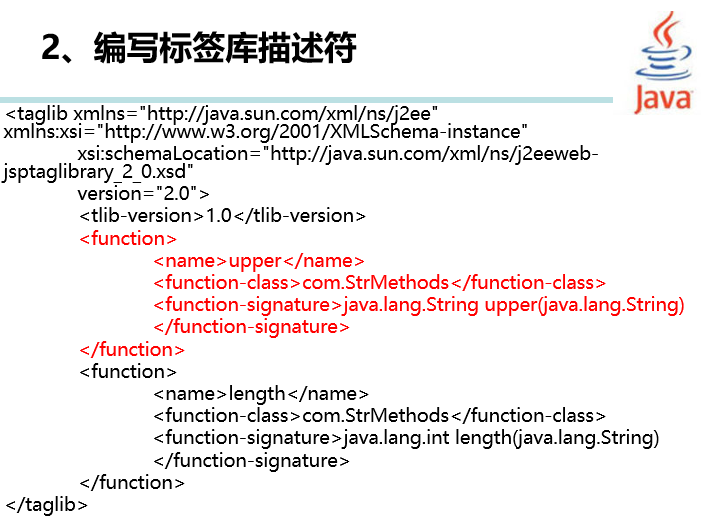
cookie header headerValues initParam pageContext

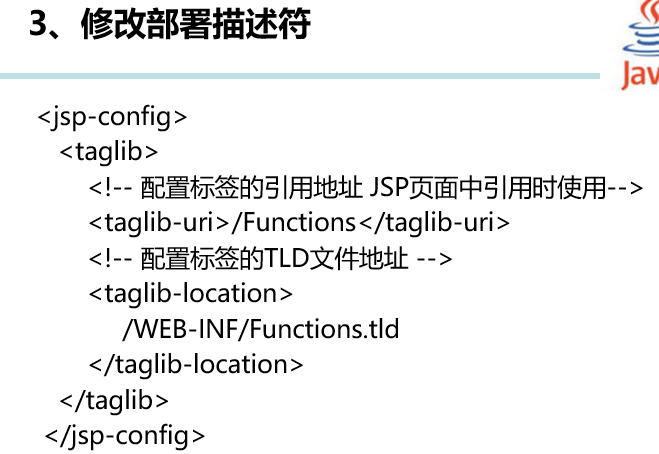
属性访问运算符和集合访问运算符

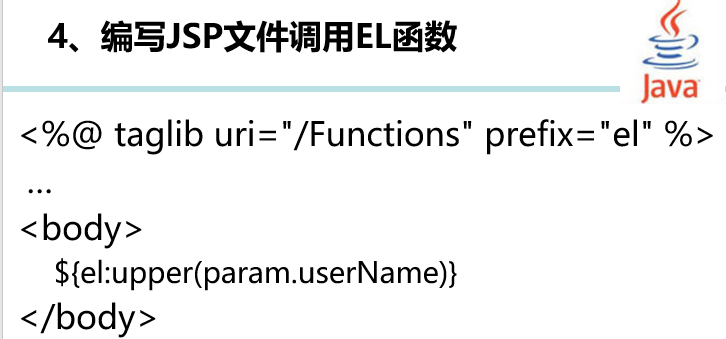
自定义EL函数

1. 编写方法所在源文件.java文件
2. 编写标签库描述符.tld文件
3. 修改部署描述符web.xml文件
4. 编写JSP文件









<http://www.edu2act.cn/team/2017-ji-Java-qi-ye-ji-ying-yong-kai-fa-er/tasks/rosegraph>

自定义函数与自定义标签需加强。

Servlet规范定义的4种认证机制

HTTP基本认证

浏览器发送一个访问受保护资源的请求

服务器观察到请求的资源是受保护的，于是服务器发送一个代表未授权的401代码到客户端

在接收到上述相应后，浏览器会打开一个对话框，提示用户输入用户名和密码，用户输入了用户名和密码，浏览器再次发送请求，并在头信息中传递值。服务器接收该请求，检验用户名和密码，如果合法则发送请求资源给客户端，否则再起发送401未授权代码

非常容易构建，所有浏览器均支持

安全性不能保证，无法定制与应用程序向匹配的对话框外观

HTTP摘要认证

在基本认证的基础上密码是加密发送，比HTTP基本认证安全

但是仅由微软的IE浏览器支持，因为规范没有对HTTP摘要认证强制，许多Servlet容器不提供支持

HTTPS客户端认证

构建在安全套接字之上的HTTP

是四种认证中最安全的，被所有浏览器支持

需要认证中心颁发的认证书

实现、维护该认证具有较高的成本

HTTP表单认证

非常容易构建，所有浏览器均支持，可以定制登录窗体的外观

不安全，因为用户名和密码未加密，只有在Cookie维持会话时，才可以使用。