# JSP

JSP（全称Java Server Pages）是由 Sun Microsystems 公司倡导和许多公司参与共同创建的一种使软件开发者可以响应客户端请求，而动态生成 HTML、XML 或其他格式文档的Web网页的技术标准。

JSP环境：JDK和Tomcate服务器

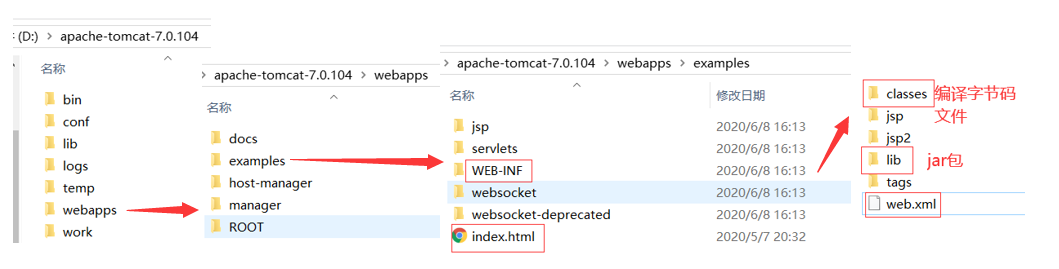
## Tomcate服务器

### Tomcate目录结构

[参考](https://www.cnblogs.com/yadongliang/p/6806920.html)



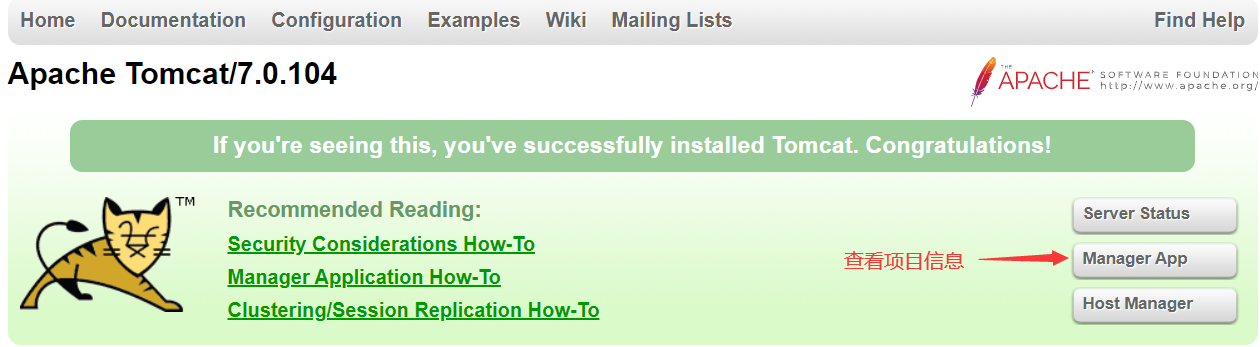
/webapps子目录：



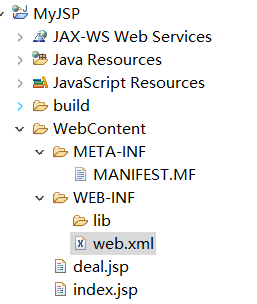
### 启动Tomcate

bin/startup.bat启动🡪浏览器<http://localhost:8080/>

账号密码需要配置conf\tomcat-users.xml



### JSP项目结构



JSP文件格式：

<%@ page language="java" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>JSP首页</title>

</head>

<body>

</body>

</html>

xml文档：

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"

version="3.0" metadata-complete="true">

<display-name></display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.xhtml</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

<session-config>

<session-timeout>20</session-timeout>

</session-config>

</web-app>

### 修改服务器端口号

conf/server.xml：

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

### 解决url乱码

conf/server.xml：

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" URIEncoding="utf-8"/>

## JSP语法

<%@ page language="java" import="java.util.\*" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="keywords" content="k1,k2,k3"/>

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

<title>JSP首页</title>

</head>

<body>

</body>

</html>

### 开头

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

### 注释

<!-- 我是注释，客户端可见 -->

<%-- 我是注释，客户端不可见--%>

<%

//单行注释

/\*

多行注释

\*/

%>

### 脚本

<%

//输出

out.println("输出内容");

%>

<%!

//定义变量、方法

String str = "我是字符串";

int add(int a,int b){

return a+b;

}

%>

<!-- 使用变量、方法 -->

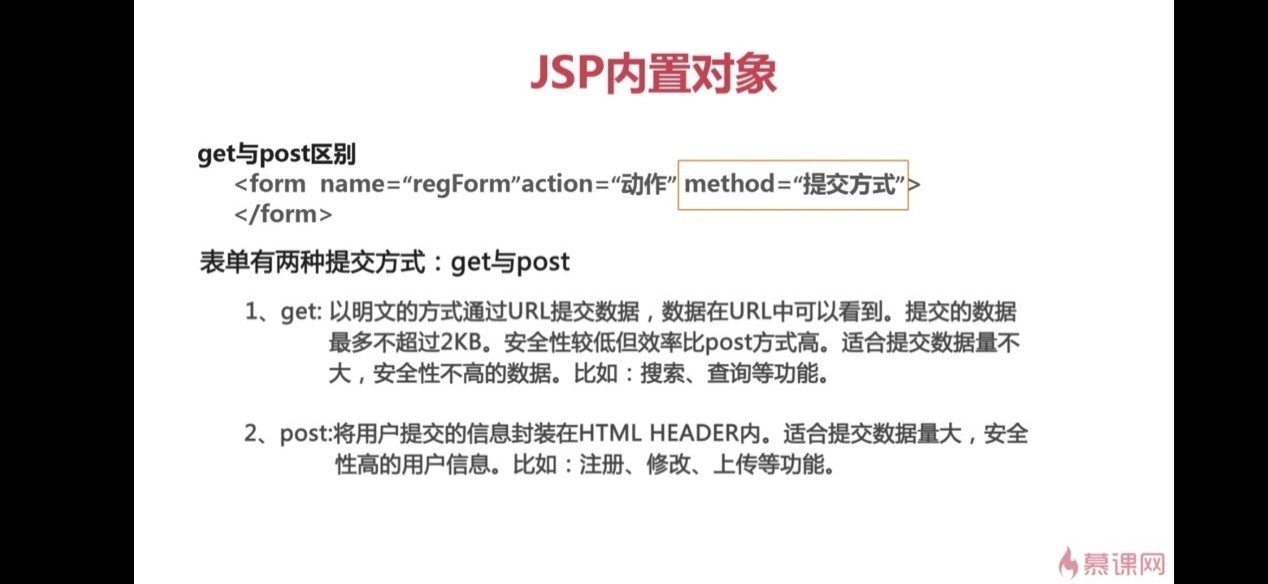
<%=str %><br>

<%=add(2,3) %><br>

## JSP内置对象

**九大内置对象**：out、request、response、session、application、【Page、pageContext、exception、config】

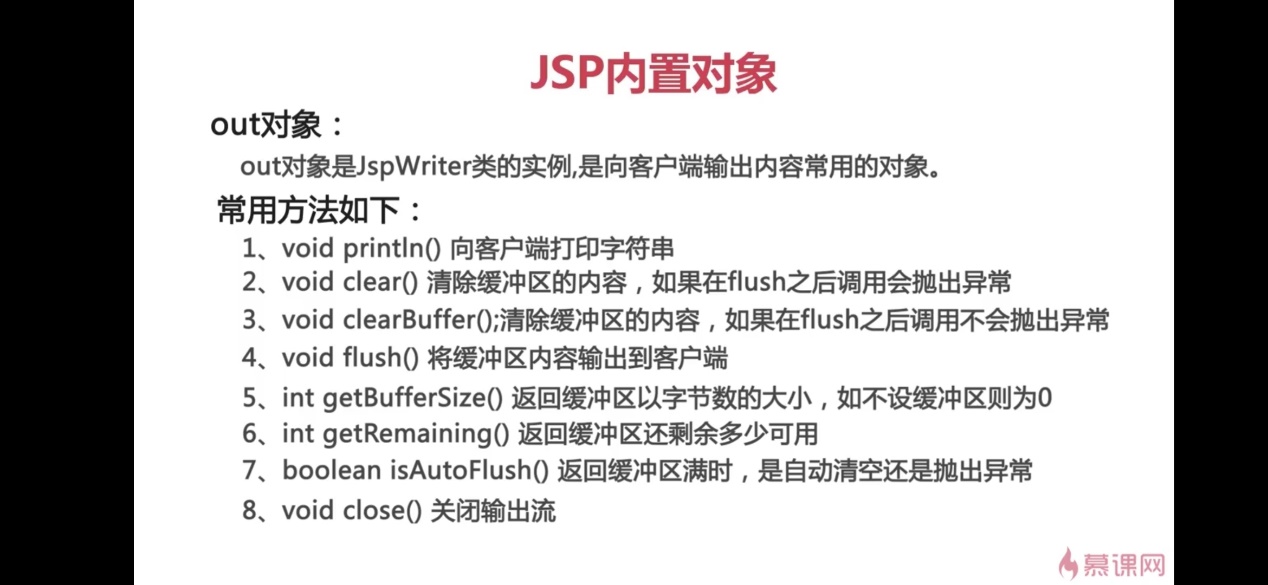
**表单补充：**



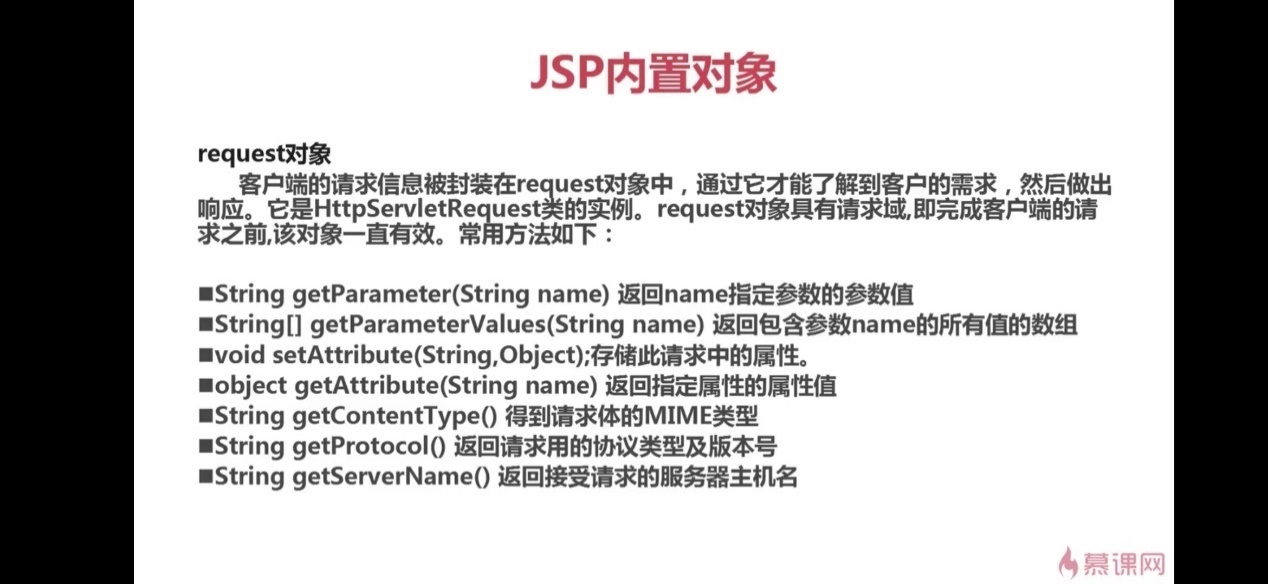
url传参数：

<a href="deal.jsp?sex=man&age=28">链接</a>

### out



### request



index.jsp：

<form name="myForm" action="deal.jsp" method="post">

<table>

<tr>

<td>用户名</td>

<td><input type="text" name="username"></td>

</tr>

<tr>

<td>密码</td>

<td><input type="password" name="password"></td>

</tr>

<tr>

<td colspan="2"><input type="submit" value="提交"></input></td>

</tr>

</table>

</form>

deal.jsp：

<%

request.setCharacterEncoding("utf-8");//解决form中文乱码，无法解决url传参

%>

<%

String username = request.getParameter("username");

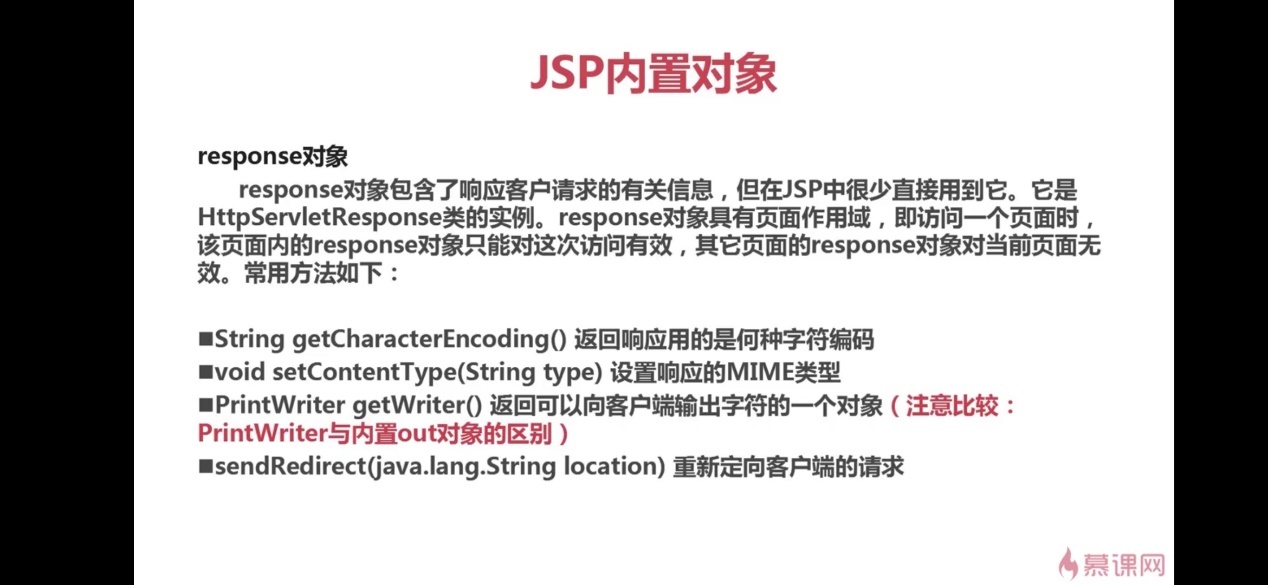
String password = request.getParameter("password");

%>

<%=username %>

<%=password %>

### response



<%

response.setContentType("text/html;charset=utf-8");

PrintWriter outer = response.getWriter();

outer.print("输出内容");//注意：使用response的printWriter打印的内容在out内置对象打印内容之前

//解决办法：对out对象使用flush方法

%>

#### 请求转发和重定向

转发是在服务器端进行，重定向是在客户端进行，因此，重定向地址会反映在浏览器地址栏上。同时，重定向相当于浏览器两次请求，因此，前一次请求的对象不会保存。

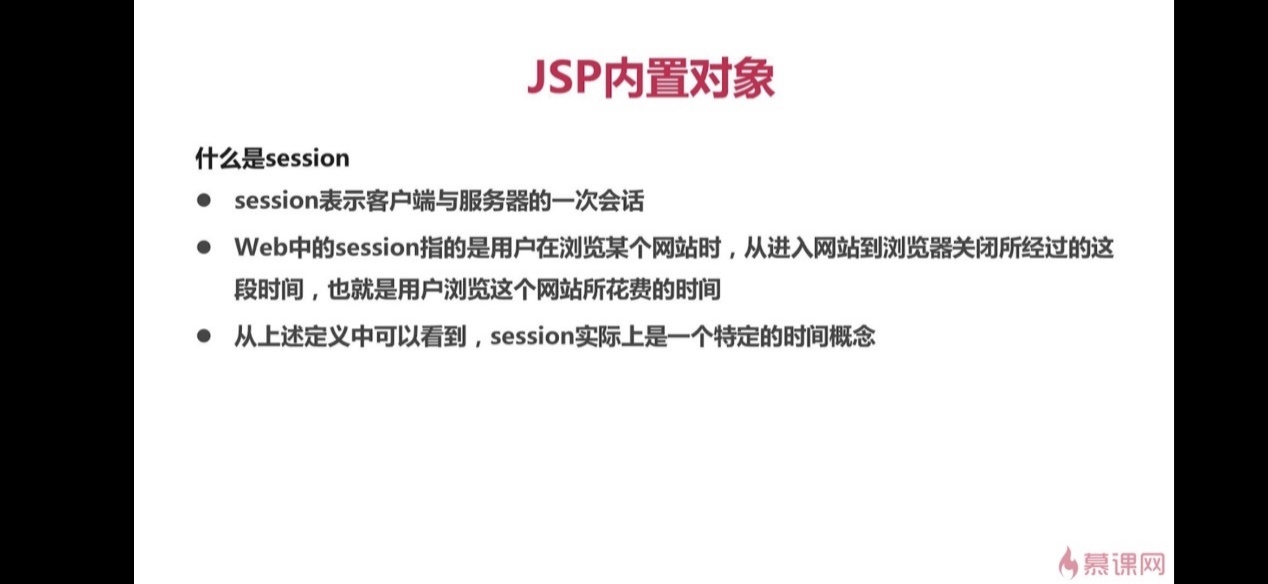
<%

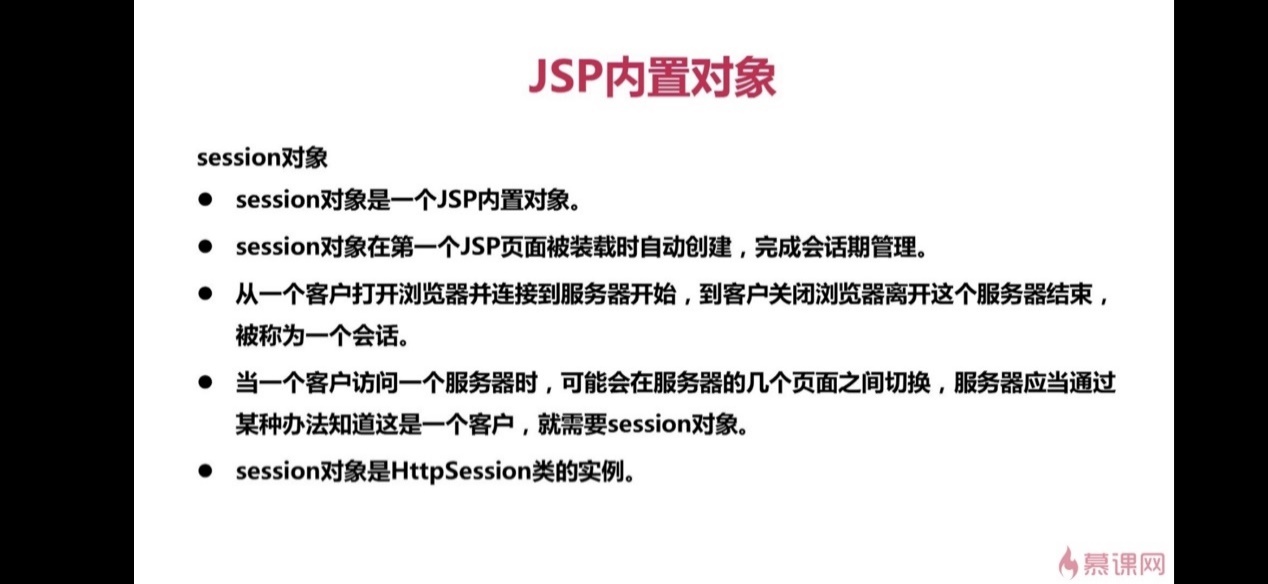
response.sendRedirect("index.jsp");//请求重定向

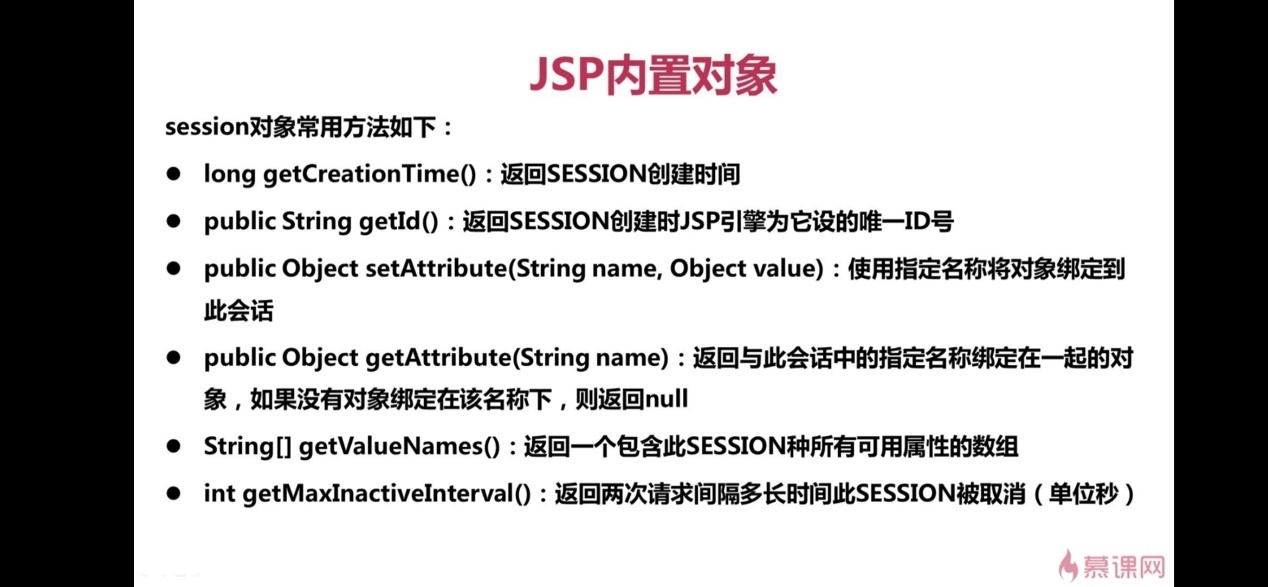
request.getRequestDispatcher("index.jsp");//请求转发

%>

### session







#### session的生命周期

**创建**🡪打开指定页面

**活动**🡪通过超链接打开新页面属于同一次会话；同一页面同时打开多次属于同一次会话

**销毁**🡪调用session.invalidate()方法；session过期；服务器重启

#### 设置session生命周期

* Tomcate默认session为30分钟
* 使用session.setMaxInactiveInterval//单位秒
* 在web.xml中配置//单位分钟

<session-config>

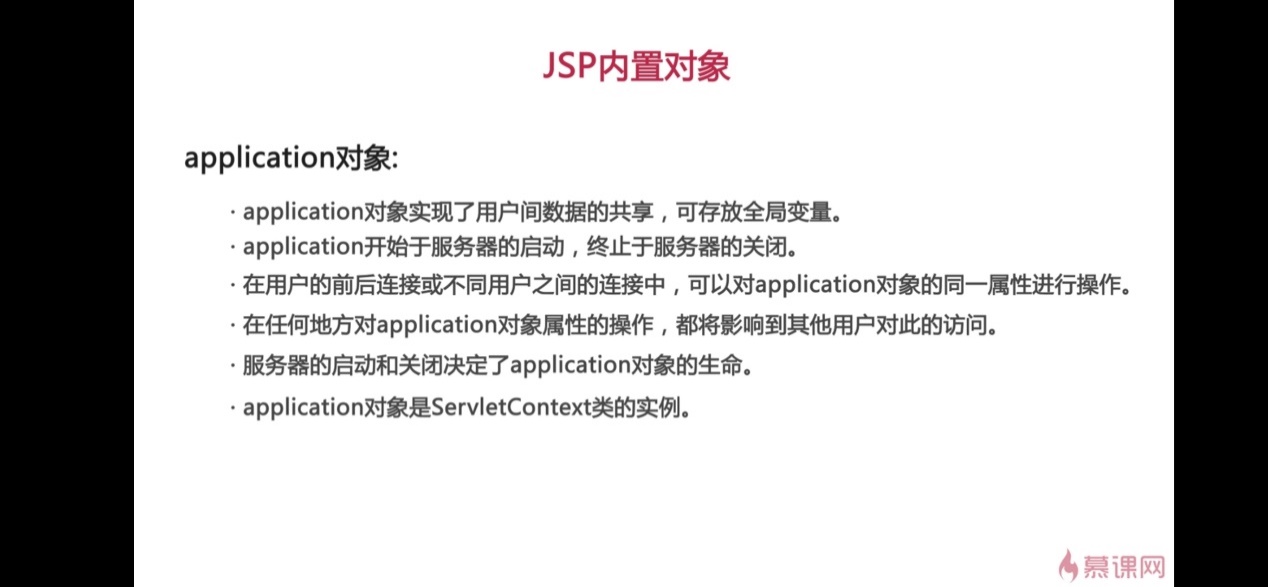
<session-timeout>

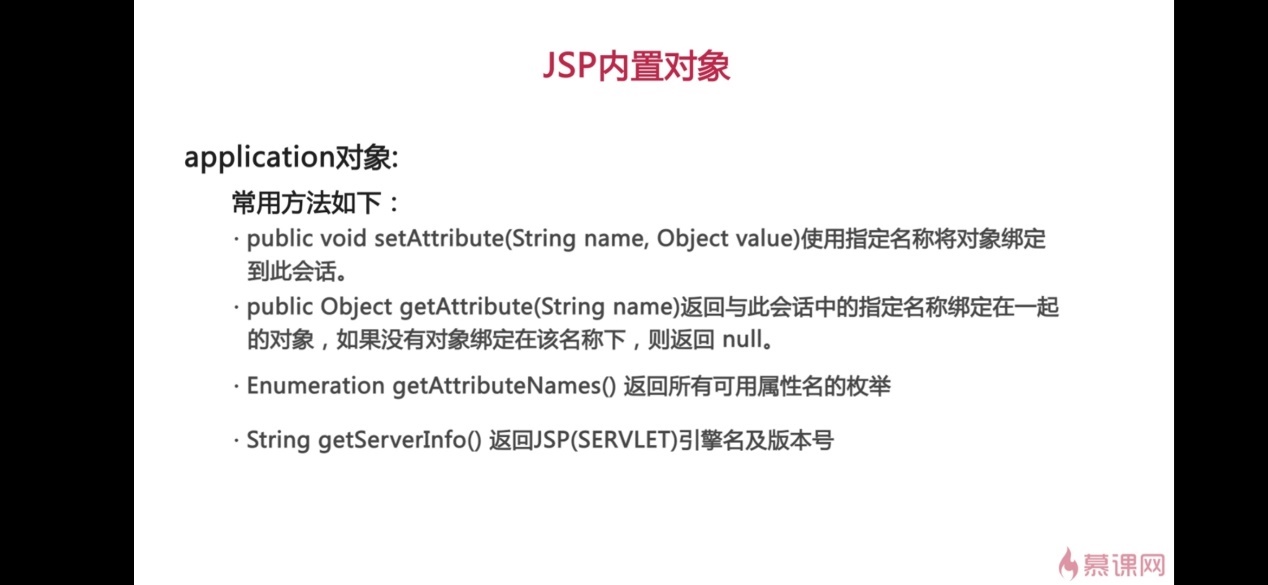
10

<session-timeout>

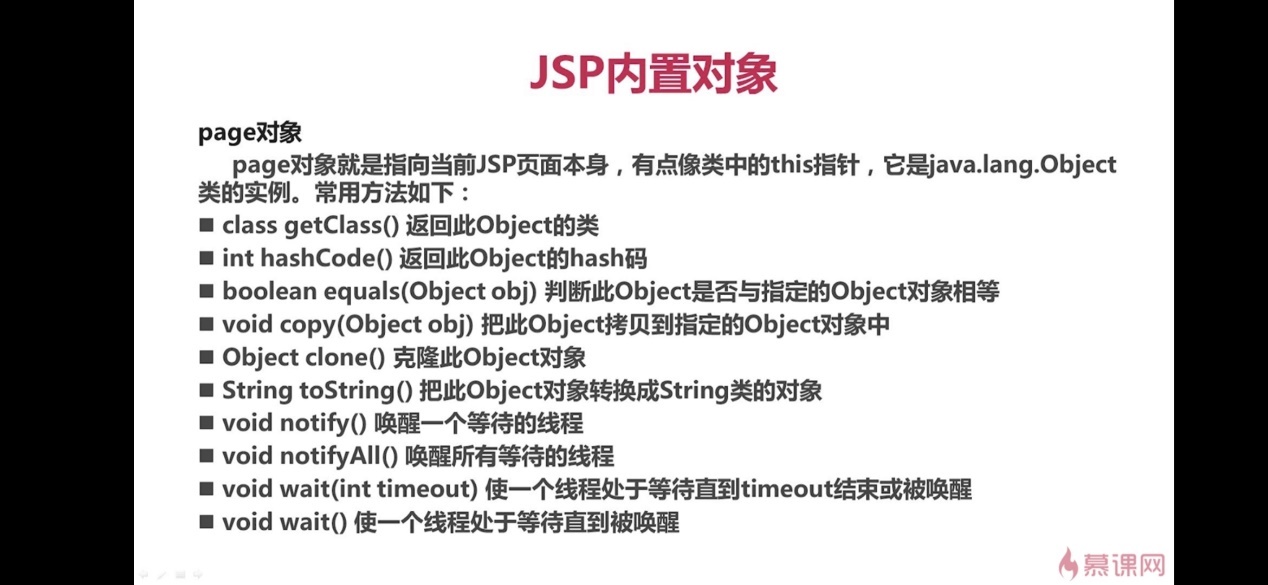
</session-config>

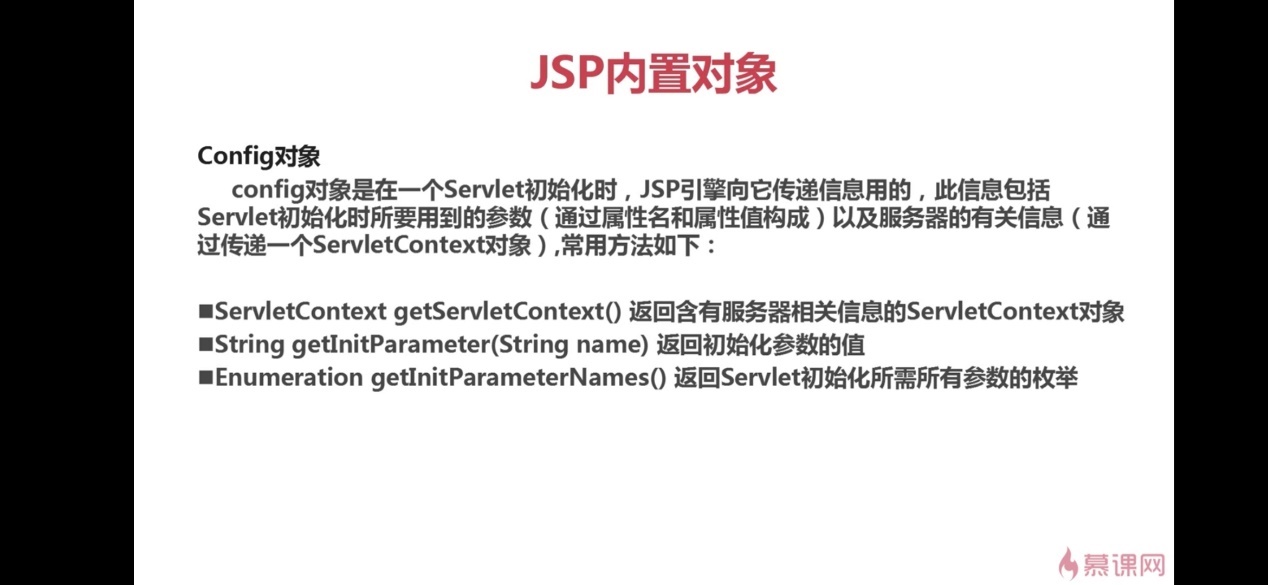
### application

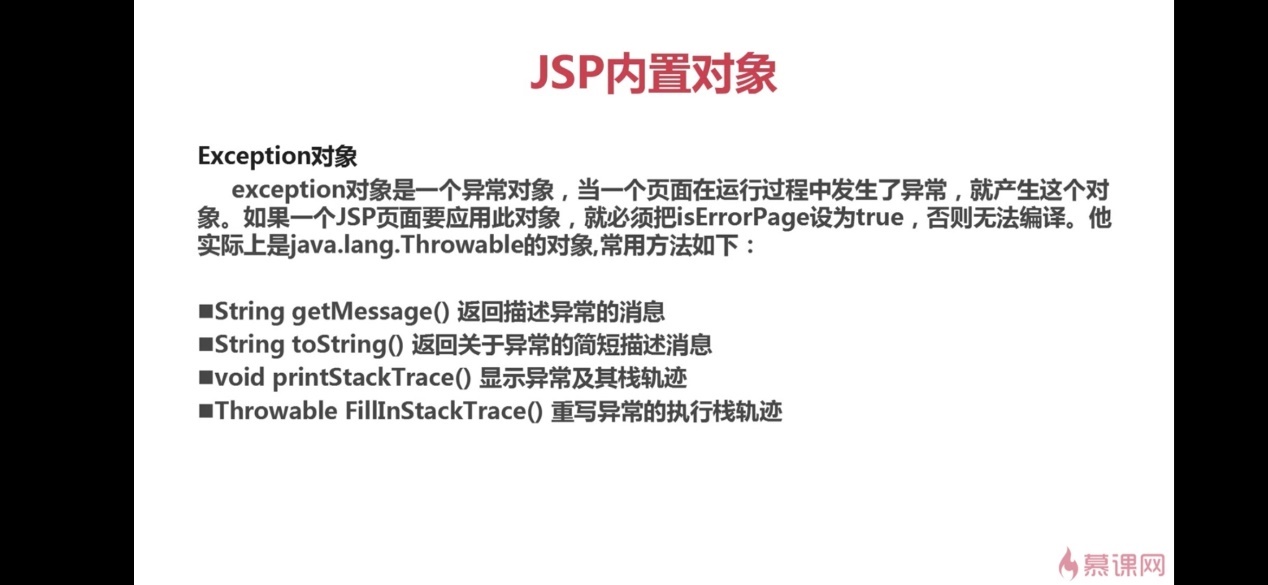




### 其他





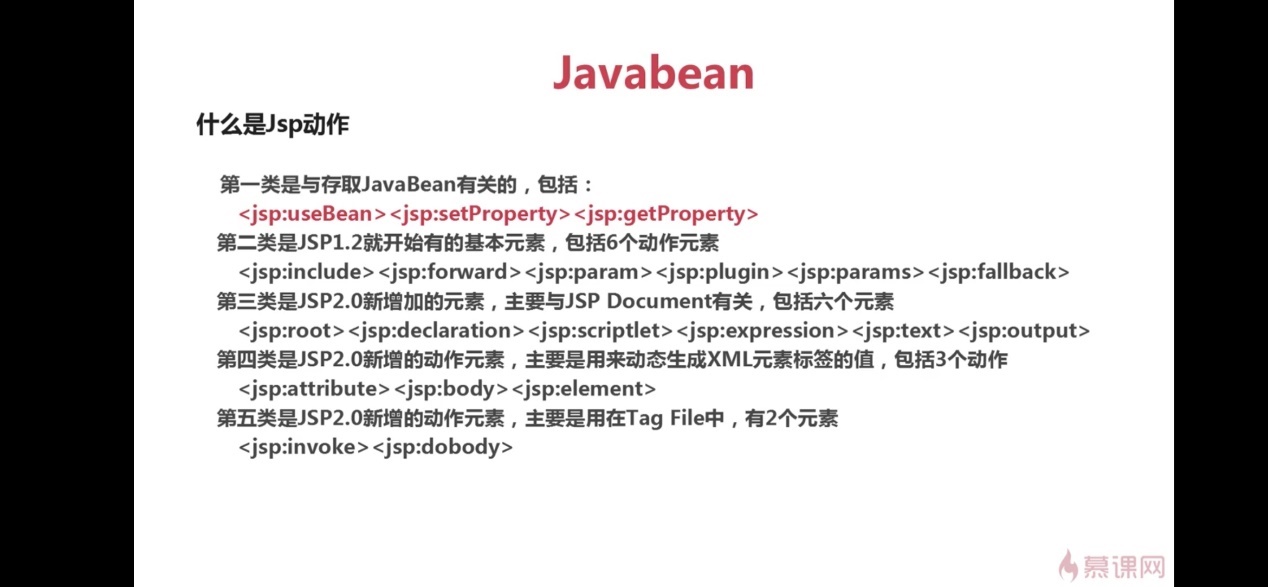


## JSP动作

### 定义

动作元素为请求处理阶段提供信息，符合xml语法的语句

### 相关标签



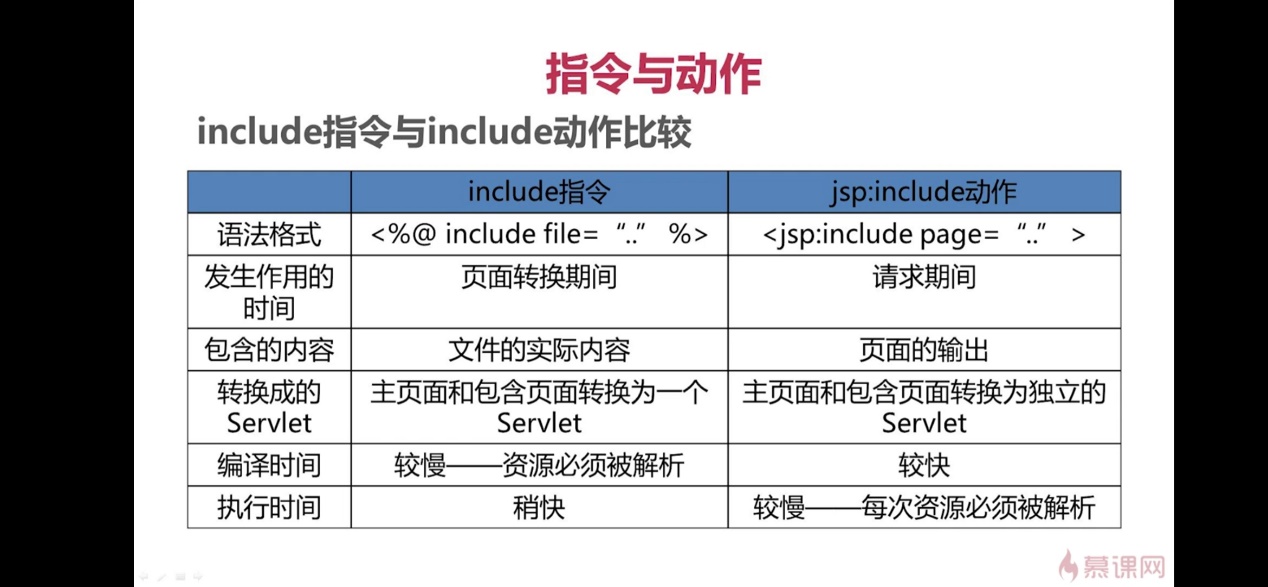
#### include

<!-- include指令:引用别的jsp页面 -->

<%@ include file="deal.jsp" %>

<!-- include动作:page-包含的页面;flush-是否从缓冲区读取 -->

<jsp:include page="deal.jsp" flush="false"/>



#### forward

<!-- 页面转发 -->

<jsp:forward page=""/>

<!-- 等同于request.getRequestDispatcher("URL").forward(request,response) -->

#### param

<!-- param常作为forward的子标签 -->

<jsp:forward page="deal.jsp">

<jsp:param value="28" name="age"/>

<jsp:param value="xiangyang" name="address"/>

</jsp:forward>

## JavaBean

### 设计原则

共有类public、有无参构造方法、属性私有且有set和get方法

### 创建javaBean

public class Student {

private String name;

private String id;

public Student() {

super();

}

public Student(String name, String id) {

super();

this.name = name;

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

}

#### 普通方式

<%

Student stu = new Student();

stu.setId("001");

stu.setName("张三");

%>

<%=stu.getName() %>

#### 使用jsp动作标签

<jsp:useBean id="MyStudent" class="ccnu.Student" scope="page"/>

<%=MyStudent.getName() %>

<form name="myForm" action="deal.jsp" method="post">

<table>

<tr><td><input type="text" name="name"></td></tr>

<tr><td><input type="text" name="id"></td></tr>

<tr><td><input type="text" name="myid"></td></tr>

</table>

</form>

<!-- 赋值方式1：表单自动匹配所有元素，要求表单name要和类属性同名 -->

<jsp:setProperty property="MyStudent" name="\*"/>

<!-- 赋值方式2：表单部分匹配元素 -->

<jsp:setProperty property="MyStudent" name="name"/>

<!-- 赋值方式3：手工赋值 -->

<jsp:setProperty property="MyStudent" name="id" value="001"/>

<!-- 赋值方式4：与request参数关联 -->

<jsp:setProperty property="MyStudent" name="id" param="myid"/>

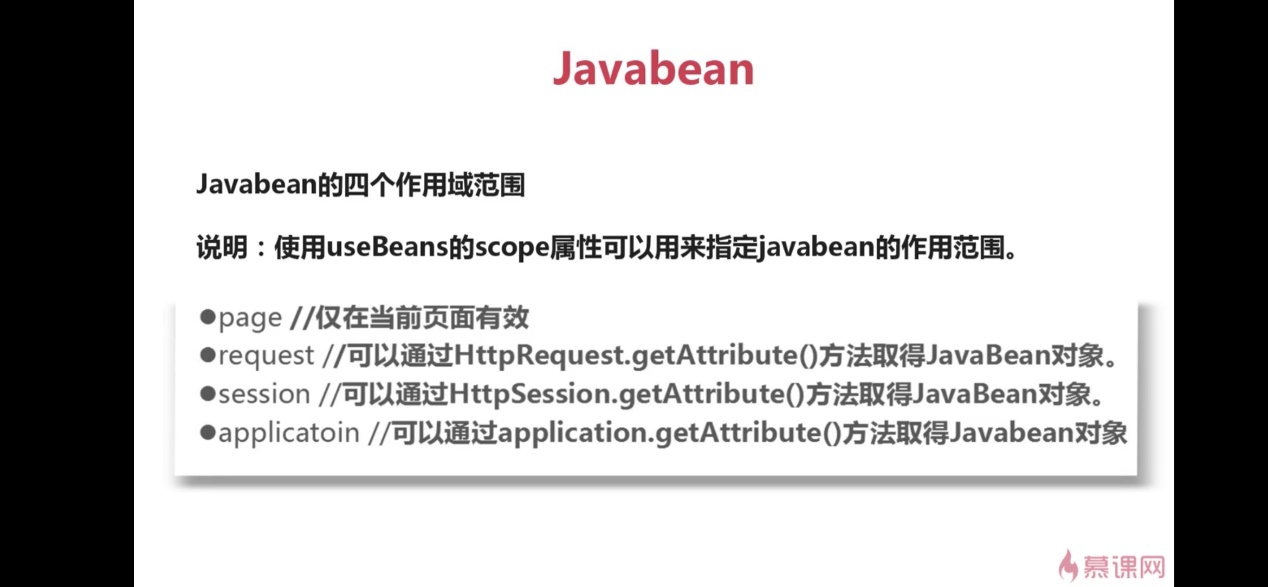
<!-- 获取值 -->

<jsp:getProperty property="MyStudent" name="id"/>

### javaBean的作用范围

<jsp:useBean id="MyStudent" class="ccnu.Student" scope="page"/>

scope范围：



## cookie

保存用户状态的两种机制：session、cookie

cookie的作用：保存浏览记录、简化登录、对特定对象追踪

### cookie使用





注意：

* + 在使用setMaxAge方法后要调用addCookie方法进行保存
  + 乱码问题，使用java.net.\*的URLEncoder类进行转码，使用时用URLDecoder进行解码

### session和cookie的区别

