JSP本质上是一个servlet，是一种动态网页技术，实现了网页的处理逻辑与显示的分离

我们在jsp中书写java，html，css等代码，在服务器端jsp引擎读取jsp文件，解释jsp脚本，生成HTML页面并返回给浏览器，也就是说服务器端执行jspservlet把.jsp文件转译成servlet文件，再执行转译后的servlet

javaBean

必须有public的无参的构造器

如果成员变量提供了getter/setter方法，那么称之为javabean属性。

JavaBean主要是用来通过反射操作的类！

因为需要通过Class的newInstance()方法来创建类的实例，所以要求类必须提供public的无参构造器

Servlet进行请求的逻辑处理，jsp进行页面的展现

# Jsp三种注释

前端语言注释

Java语言注释

Jsp注释 <%-- --%>

# Page指令

<%@page 属性名=“属性值” 属性名=“属性值”... %>

Language:声明jsp要被转译的语言

Import：导入的包、

pageEncoding:设置jsp文件的数据编码格式

session：设置转译的servlet是否开启session支持，默认开启

errorPage：设置jsp运行错误跳转的页面

extends：设置jsp转译的java文件要继承的父类（包名+类名）

作用：

配置jsp转译的相关参数

# Jsp的静态引入和动态引入

静态引入：合并jsp，即将当前jsp和静态引入的jsp文件转译成一个servlet文件，因此静态引入不支持同名变量

<%@include file=”文件路径” %>

动态引入：调用jsp，即将引入的jsp单独转译成servlet文件，在当前转译好的sevlet文件中调用动态引入的servlet文件，因此动态引入支持同名变量

<jsp:include page=”文件路径”></jsp:include>

# Forward转发标签

<jsp:forward page=”文件路径”></jsp:forward>

在转发标签中只可加入如下标签

<jsp:param name=”…” value=”…” />

会将数据以？分割拼接在文件路径后边

# 九大内置对象

——jsp文件在转译成servlet文件时，自动生成并声明的对象。

内置对象只能在局部代码块或脚本段中使用，不能在全局代码块中使用。

**PageContext:**页面上下文对象，封存了其他内置对象，即当前jsp的运行信息

作用域：当前页面

**Request：**封存当前请求数据的对象，由tomcat服务器创建，一次请求

**Response：**响应对象，用来将处理结果响应给浏览器的对象，设置响应头，重定向

**Session；**用来存储用户的不同请求的共享数据，一次会话

**Application：**即ServletContext对象，一个项目只有一个，存储用户共享数据的对象。项目内

**Out：**响应对象，jsp内部使用，带有缓冲区的对象，效率高于response对象

**Page：**代表当前jsp对象，相当于this

**Exception：**异常对象

使用此对象需要在page指定中使用属性isErrorPage=“true”开启

**Config： 即servletConfig，用来获取web.xml中的配置数据，完成一些初始化数据的读取**

**四个作用域对象：**

PageContext:当前页面，解决了当前页面内的数据共享问题，获取其他内置对象

Session：一次会话，

解决了一个用户不同请求的数据共享问题

将数据从一次请求流转给其他请求

Request：一次请求

解决了一个请求内Servlet的数据共享问题

通过请求转发，将数据流转给下一个servlet

Application：项目内，

解决了不同用户的数据共享问题

将数据从一个用户流转给另一个用户

作用：数据流转

# Jsp路径

在jsp中资源路径可以使用相对路径完成资源的跳转，但是：

缺陷：资源位置不可随意更改，

需要使用../进行 跳转，比较繁琐

**使用绝对路径**：（开发常用）

/虚拟项目名/项目资源路径

绝对路径中第一个/代表tomcat服务器根目录，即localhost:8080

使用jsp自带的全局路径声明：

作用：

给资源添加项目路径：localhost:8080/虚拟项目名

# Taglib指令