什么是域对象？

**存在作用域的对象就是域对象。域对象在javaWeb开发中的地位贼高，因为在域对象中可以实现数据共享！**

**在此对象的作用域内，实现数据共享！**

域对象操作数据的方法

**setAttribute(“数据关键字”, 值) 往域对象中添加数据的方法，如果域对象中已经存在了此数据关键字，则为修改数据，值是object类型，可以存放各种类型的数据！**

**getAttribute(“数据关键字”) 获取域对象中的数据**

**removeAttribute(“数据关键字”) 删除域对象中的数据**

request域对象与ServletContext域对象的区别

**request域对象：**

**请求到达servlet时会创建request域对象**

**请求完成、做出响应销毁本次请求**

**作用范围是一次请求**

**+ServletContext域对象：**

**服务器加载项目时创建**

**服务器正常关闭销毁**

**作用范围整个web项目**

jsp与servlet的区别

**作用：jsp趋向于视图层(展示数据，请求发起)，**

**servlet趋向于控制层(接收请求、响应数据、页面跳转、方法调用)**

**文件本质：jsp文件、纯粹的java类**

**内置对象：jsp九大内置对象，servlet没有**

请求转发与重定向的区别：（重点）

**(1) 、请求转发是服务器行为，而重定向是客户端浏览器行为**

**(2) 、请求转发的速度要快于重定向**

**(3) 、转发是一次请求、一次响应，而重定向是多次请求、多次响应**

**(4) 、转发的url地址栏不变，而重定向会发生变化**

**(5) 、转发是在同一台服务器上完成的，只能访问服务器内部资源，但是重定向可以访问网站**

get与post的区别：

**get:**

**长度：当发送数据时，get方法向url中添加数据；URL的长度是受限制的（URL的最大长度是2048）**

**数据类型：只允许ascll字符**

**安全性：差（因为发送的数据是url的一部分）**

**可见性：对所有人可见。所以后退键无害**

**编码类型：：表单enctype属性：application/x-www-form-urlencoded///**

**post:**

**长度：不限制**

**数据类型：任意类型**

**安全性：高，因为参数不会保存在浏览器历史或者服务器的日志中**

**可见性：数据不会显示在url中**

**编码类型：：表单enctype属性：application/x-www-form-urlencoded///或者multipart/form-data**

**九大内置对象：**

**application(ServletContext) 代表全局管理者对象，服务于整个web项目**

**config(ServletConfig) 获取当前servlet中的初始化参数**

**out(JspWriter) 主要用于输出内容，比如在jsp编译后的java文件中，页面就是由此对象响应出去的！**

**page(Object) 代指当前页面**

**pageContext(PageContext) 当前页面的管理者对象**

**request(HttpServletRequest) 请求**

**response(HttpServletResponse) 响应**

**session(HttpSession) 表示会话对象**

**exception 异常对象**

动态包含和静态包含的区别

1、使用include指令包含称为静态包含，使用include动作标签包含称为动态包含

2、静态包含文件路径书写在指令的file属性中，而动态包含文件路径书写在标签的page属性中

3、静态包含被包含的文件不会单独编译解析，而是直接将内容导入到源文件中一并进行编译与解析！而动态包含被包含的文件也会单独进行编译解析！

4、如果想要对被包含的文件进行处理，那么首选动态包含，因为有源文件。所以一般情况下，静态包含用于包含html静态文件，动态包含用于包含jsp动态文件！

**方式：静态指令、动态动作标签**

**属性：静态file、动态page**

**文件类型：静态html、动态jsp**

**编译方式：静态不单独编译、动态单独编译**

jsp的动作标签：

jsp的基础语法

jsp的三大指令：<%@ 指令名称 属性名=”属性值” ... %>

jsp的九大内置对象

jsp的四大域对象

**语法格式：<jsp:动作标签名 属性名=”属性值” ...></jsp:动作标签名>**

**支持请求转发：forward 转发地址书写在page属性中**

**支持页面包含：include 被包含的文件名写在page属性中**

Session与Cookie的区别？

**1、Session存储于服务器、而Cookie存储于客户端浏览器**

**2、Session的生命周期：默认30分钟超时、服务器正常关闭、执行了invalidate()方法，Cookie默认浏览器会话结束，但是可以通过setMaxAge()来设置存活时间**

**3、Session中存储数据大小无限制，而Cookie存储只能到4kb**

**4、Session中可以存储任意类型的数据，而Cookie只能存储字符串文本信息**

**5、Session存储在服务端安全性更高，而Cookie存储在浏览器客户端相对来说安全性较低(一般存储的数据都会进行加密)**

**备考**

**jstl标准标签库(Java server pages standarded tag library)**

**组成：core、格式化、数据库、xml、函数库**

**使用方式：**

**先导入jar包：jstl、standard**

**引入库文件：taglib指令 prefix起别名，uri找到路径**

**core核心标签库：**

**c:if test条件判断**

**c:forEach循环遍历**

**固定次数 begin开始 end结束 varStatus循环状态**

**数组、集合联用**

**items=”${stu}”表示要遍历的集合**

**var每一个遍历元素的别名**

**varStatus循环状态**

**c:set var（变量的名称） value（变量的值）**

**目前在页面中的输出方式：**

1. **c:out value**
2. **el表达式**
3. **输出响应流（也就是out.write）**
4. **脚本表达式**

**c:redirect url：重定向的地址**

**<c:choose> //多重判断**

**c:when test=“条件”**

**c:otherwise default条件全部不满足**

**</c:choose>**

jsp的指令：

**page指令：描述页面信息**

**三大必备属性：**

**language contentType pageEncoding**

**三大可选属性：**

**errorPage session import**

**include指令：实现页面包含**

**file=外界文件路径名（whtaever.jsp）**

**注意事项：**

**(1) 、include指令可以出现在jsp文件的任意位置，什么时候想要包含都可以，page指令其实也可以出现在任何位置，但是一般都会将page指令书写在页面的最上方。**

**(2) 、include指令也可以出现多次**

**(3) 、引入的外界文件路径书写在include指令的file属性中**

**(4) 、使用include指令来包含文件的方式我们称为静态包含，被包含的文件不会单独解析与编译，而是将文件内容整体导入到源文件中，统一对源文件进行解析与编译！**

**taglib指令：引入第三方标签库**

**prefix：为引入的标签库中的标签设置一个前缀别名**

**uri：引入的标签库路径、位置**

**使用el表达式获取域对象中数据的语法格式为：**

**${域对象中数据存储的关键字 }**

**如果关键字对应的是一个对象，获取对象属性值的语法为:**

**${对象存储的关键字.属性名 }**

**关于session和cookie的补充：**

**Cookie （储存在用户本地终端上的数据,在javaWeb开发中，本地终端就是浏览器）**

**Cookie的注意事项：**

**Cookie中只能存储(轻量级、简化版、少量的)字符串数据，最大支持到4kb**

**如何获取Session？**

**1、jsp中，session是内置对象，直接使用即可**

**2、servlet中，获取session的代码为request.getSession() ==> HttpSession对象**