web

1. Servlet

1.1. 生命周期

1.1.1. Servlet 加载—>实例化—>服务—>销毁。

init（）：

在Servlet的生命周期中，仅执行一次init()方法。它是在服务器装入Servlet时执行的，负责初始化Servlet对象。

service（）：

它是Servlet的核心，负责响应客户的请求。每当一个客户请求一个HttpServlet对象，该对象的Service()方法就要调用，而且传递给这个方法一个“请求”（ServletRequest）对象和一个“响应”（ServletResponse）对象作为参数。在HttpServlet中已存在Service()方法。默认的服务功能是调用与HTTP请求的方法相应的do功能。

destroy（）：

仅执行一次，在服务器端停止且卸载Servlet时执行该方法。当Servlet对象退出生命周期时，负责释放占用的资源。

1.2. 重定向和转发

1.2.1. response.sendRedirect("index.jsp"); 不能携带数据

1.2.2. request.getRequestDispatcher("index.jsp").forward(request, response);能携带数据

1.3. session

1.3.1. Session和Cookie的区别

Cookie是把用户的数据写给用户的浏览器；

Session技术把用户的数据写到用户独占的session中（即保存在服务器端）

1.3.2. HttpSession session = request.getSession();

1.3.3. 在Java中，默认session在30分钟之内未被使用，服务器就会将session销毁，不管用户浏览器此时有没有关闭。即便用户关闭了浏览器，结束了当前会话，session也不会立即销毁，直到默认时效内session未被使用。

手动销毁session: session.invalidate();

2. MVC

2.1. 模型(model)－视图(view)－控制器(controller)

2.2. 分层，结构清晰，耦合性低，大型项目代码的复用性得到极大的提高，开发人员分工明确，提高了开发的效率，维护方便，降低了维护成本。

3. JSP

3.1. 内置对象

3.1.1. request、response、session、application、

out、pagecontext、config、page、exception

request

request对象代表了客户端的请求信息，主要用于接受通过HTTP协议传送到服务器的数据。（包括头信息、系统信息、请求方式以及请求参数等）。request对 象的作用域为一次请求。

getParameter(String strTextName) 获取表单提交的信息.

getServletPath() 获取客户提交信息的页面。

getMethod() 获取客户提交信息的方式

setCharacterEncoding("UTF-8"); 设置统一编码

session

是一个**JSP**内置对象，它在第一个**JSP**页面被装载时自动创建，完成会话期管理。从一个客户打开浏览器并连接到服务器开始，到客户关闭浏览器离开这个服务器结束，被称为一个会话。

public void setAttribute(String key,Object obj)：将参数**Object**指定的对象**obj**添加到**Session**对象中，并为添加的对象指定一个索引关键字。

public Object getAttribute(String key)：获取**Session**对象中含有关键字的对象。

public String getId()：获取**Session**对象编号。

application

application对象可将信息保存在服务器中，直到服务器关闭，否则**application**对象中保存的信息会在整个应用中都有效。与**session**对象相比，**application**对象生命周期更长，类似于系统的“全局变量”。

setAttribute(String key,Object obj)：将参数**Object**指定的对象**obj**添加到**Application**对象中，并为添加的对象指定一个索引关键字。

getAttribute(String key)：获取**Application**对象中含有关键字的对象。

pagecontext

pageContext 对象的作用是取得任何范围的参数，通过它可以获取**JSP**页面的**out**、**request**、**reponse**、**session**、**application** 等对象。

response

response

代表的是对客户端的响应，主要是将JSP容器处理过的对象传回到客户端。

response对象也具有作用域，它只在JSP页面内有效。

response.sendRedirect(index.jsp);重定向

out

out

对象用于在Web浏览器内输出信息，并且管理应用服务器上的输出缓冲区。在使用 out

对象输出数据时，可以对数据缓冲区进行操作，及时清除缓冲区中的残余数据，为其他的输出让出缓冲空间。待数据输出完毕后，要及时关闭输出流。

out.print()：输出各种类型数据。

out.newLine()：输出一个换行符。

out.close()：关闭流。

page

page 对象代表JSP本身，只有在JSP页面内才是合法的。

page隐含对象本质上包含当前

Servlet接口引用的变量，类似于Java编程中的 this 指针。

config

config 对象的主要作用是取得服务器的配置信息。通过 pageConext对象的 getServletConfig() 方法可以获取一个config对象。当一个Servlet

初始化时，容器把某些信息通过 config对象传递给这个

Servlet。开发者可以在web.xml

文件中为应用程序环境中的Servlet程序和JSP页面提供初始化参数。

exception

exception 对象的作用是显示异常信息，只有在包含 isErrorPage=”true” 的页面中才可以被使用，在一般的JSP页面中使用该对象将无法编译JSP文件。

3.1.2. 请求转发和重定向

1. 地址不一样

2. 请求次数不一样

请求转发 一次

重定向 两次

3. 数据无法传递的问题

重定向无法保存request中的数据

4. 效率问题

请求转发客户端只请求一次，因此效率较高

5. 跳转范围有限制

请求转发： 只能针对当前的项目

重定向： 没有任何限制

3.2. JSP语法

3.2.1. 脚本程序的语法格式:<% 代码片段 %>

3.2.2. JSP声明:<%! declaration; [declaration;]+ ... %>

3.2.3. JSP表达式:<%= 表达式 %>

表达式元素中可以包含任何符合Java语言规范的表达式，但是不能使用分号来结束表达式。

3.2.4. JSP注释:<%-- 该部分注释在网页中不会被显示--%>

3.2.5. JSP指令

<%@ page ... %>

定义页面的依赖属性，比如脚本语言、error页面、缓存需求等等

<%@ include ... %> 包含其他文件

<%@ taglib ... %> 引入标签库的定义，可以是自定义标签

3.2.6. JSP行为

jsp:forward

从一个JSP文件向另一个文件传递一个包含用户请求的request对象

jsp:include 用于在当前页面中包含静态或动态资源

jsp:useBean 寻找和初始化一个JavaBean组件

3.2.7. JSP 表单处理

GET方法是浏览器默认传递参数的方法，一些敏感信息，如密码等建议不使用GET方法。

用get时，传输数据的大小有限制

（注意不是参数的个数有限制），最大为1024字节。

POST提交数据是不可见的，GET是通过在url里面传递的

3.3. EL表达式

3.3.1. 增加jsp代码的可读性，使脚本和HTML标签分离，使得jsp代码更易维护

3.3.2. 特点

自动转换类型

EL得到某个数据时可以自动转换类型

对于类型的限制更加的松

使用比较简单

相比较在jsp文件中嵌入java代码，EL表达式的应用更加容易

3.3.3. 语法

语法格式：

1、${EL 表达式}

2、${bean.name} or ${bean['name']}

3.3.4. 作用域（属性范围）

1、pageScope:与页面作用域(page)中的属性相关联的Map类,
主要用于获取页面范围内的属性值

2、requestScope:与请求作用域(request)中的属性相关联的Map类,主要用于获取请求范围内的属性值

3、sessionScope:与会话作用域(session)中的属性相关联的Map类,主要用于获取会话范围内的属性值

4、applicationScope:与应用程序作用域(application)中的属性相关联的Map类,
主要用于获取应用程序内的属性值

3.3.5. 运算符

1、[ ]

2、 .

作用：

1、获取对象属性

2、获取对象集合中的数据

3.4. JSTL

3.4.1. 含义：JavaServerPagesStandardTagLibrary(JSP标准标签库)

3.4.2. 库导入标签格式为：<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

3.4.3. 有时EL表达式无法实现逻辑处理，如循环、条件判断，因此仍需要java代码混合使用，JSTL可以实现逻辑控制

3.4.4. 分类

核心标签库，前缀：c

通用标签

set 作用:对作用域内容的变量或者JavaBean对象属性进行设置

<c:out value="${属性名}"></c:out>：输出标签

可以对数据进行转义输出

可以在输出时设定默认值

属性default:默认输出显示的值,如果value的值为null,则输出default的值

if标签：替代java中的if

语法：<c:iftext=”condition”var=”varName”scope=”scope”>……</c:if>

test：判断的条件

var：判断的结果

scope：判断结果存放的作用域

forEach迭代标签

语法：<c:foreachvar=”varName”items=”items”varStatus=”varStatus”>……</c:fore

作用:该标签可以替换for循环语语句,从而简化了页面中的代码,使结构更清晰,代码可读性更高

var：集合中元素的名称

items：集合对象

varStatus：当前循环的状态信息，如循环的索引号

国际化/格式化标签库，前缀：fmt

XML标签库，前缀：x

数据库标签库，前缀：sql

函数标签库，前缀：sn