1，/abc/\*.do是一个合法的映射路径。（错）

2. 创建好的Servlet，客户端可以对其进行直接访问。（错）

3. 购物车功能是基于Session实现的，所以当我们关闭浏览器时，购物车中的信息也会随之消失。（对）

4. 编写order.jsp文件时，${user.username}等价于${ user ["username "]}。（对）

5. 安装好Tomcat后，就可以直接启动运行了，并不要先安装JDK。（错）

6. 计算机中的数据都是以文本形式存储的。（错）

7. 所谓请求重定向，指的是Web服务器接受到客户端的请求后，又再次引导客户端重新发送请求，指定了一个新的资源路径。（对）

8. Tomcat容器中会话的有效时间可以在web.xml文件中设置，默认会话过期时间为30分钟。（对）

9. Tomcat容器中如果将<session-timeout>元素中的时间值设置成0或一个负数，则表示会话永不超时。（对）

10. JSP文件与html文件一样，在编写好后都可以直接在浏览器中运行（错）

11. EL是Expression Lanaguage的缩写，它是一种简单的数据访问语言，可以简化java类中代码的编写。（错）

12. EL表达式语法非常简单，都是以“${”符号开始，以“}”符号结束的。（对）

13. JSTL标签库是不需要安装的，在JSP页面中可以直接使用其中的标签（错）

14安装JSTL时需导入jstl.jar和standard.jar这两个JAR包。(对)

15. JDBC的全称是Java数据库连接，它是一套用于执行HTML语句的Java API。（错）

16. 在表单中，name属性用于指定表单的名称，以区分同一个页面中的多个表单。(对)

17. 表单提交后的数据处理程序由action属性指定（对）

18. 在HTML中，<form></form>标记用于定义表单域，即创建一个表单，以实现用户信息的收集和传递。（对）

19. <div>标记是一个区块容器标记，可以将网页分割为独立的、不同的部分，以实现网页的规划和布局。（对）

20. CSS非常灵活，既可以嵌入在HTML文档中，也可以是一个单独的外部文件。（对）

21. 大多数网页都是用HTML编写网页结构和内容，版面布局、文本或图片的显示样式则使用CSS控制。（对）

22. 在DOM文档的树形结构中，每个节点有且只有一个子节点。（错）

23. 在网页建设中，运用JavaScript可以将静态页面转变成支持用户交互并响应相应事件的动态页面。（对）

24. JavaScript不需要区分字母大小写。（错）

25. 在 HTML文档中引入JavaScript，有直接嵌入JavaScript脚本和链接外部JavaScript脚本两种。（对）

26. 在<script type="text/javascript">代码中，type属性用来指定HTML文档引用的脚本语言类型。（对）

27. window.alert()方法可以不带上window对象，直接使用alert()方法。（对）

28. 在HTTP的请求消息中，最常用的就是GET和POST方式。（对）

29. 在HTTP响应消息中，第一行为响应状态行，紧接着的是若干响应消息头，服务器端通过响应消息头向客户端传递附加信息。（对）

30. 如果在Tomcat的webapps目录下，找不到要访问的Web应用，就一定会出现找不到资源的错误。（错）

31. 在Servlet开发中，当我们要自定义一个Servlet时，必须继承GenericServlet或者【  
HttpServlet()】类

32. HttpServletResponse接口中定义的SC\_OK常量，表示状态码【200】，也就是请求成功。

33. 当传输文本时，如果编码和解码使用的码表不一致，就会导致【乱码】问题。

34. 要使当前页面定时每3秒刷新一次，可以采用HttpServletResponse中的【  
response.setHeader("Refresh","3")】方法来实现。

35. 在JSP文件中，同时存在HTML代码与【Java】代码。

36. JSP全名是Java Server Page，它是建立在【Servlet】规范之上的动态网页开发技术。

37. 新创建的JSP文件与传统的HTML文件几乎没有什么区别，唯一的区别是默认创建时，页面代码最上方多了一条【page指令】，并且该文件的后缀名是jsp，而不是html。

38. 编写Servlet类时，可以通过继承【HttpServlet】类来实现的。

39. JavaBean是Java开发语言中一个可以重复使用的软件组件，它本质上就是一个【Java类】，而且规定了在编写时要遵守相应的规范。

40. 在JavaBean中，如果一个属性既有getter方法，又有setter方法，则该属性为【读写属性】。

41. EL表达式的语法声明为【${表达式}】。

42. EL表达式中的点运算符，用于访问JSP页面中某些对象的【属性】。

43. <input />标记有多个属性，其中【type】属性为其最基本的属性，用于指定不同的控件类型。

44. <img />标签表示一个图像信息，它有一个必须要指定的【src】属性，用来指定图片路径。

45. 在CSS中，类选择器使用【英文点号】进行标识，后面紧跟类名

46. JavaScript中【单行】注释使用双斜线“//”作为注释标记。

47. 在网站建设中，HTML用于搭建页面结构，CSS用于设置页面样式，【JavaScript】用于为页面添加动态效果。

48. HTML中文译为【超文本标记语言】，主要是通过HTML标记对网页中的文本、图片、声音等内容进行描述。

49. JDBC的全称是【Java数据库连接】，它是一套用于执行SQL语句的Java API。

50. JDBC驱动管理器专门负责注册特定的JDBC驱动器，主要通过【java.sql.DriverManager】类实现。

1. 请简述MVC模式中的M、V和C各代表什么。

模型（Model）：模型代表应用程序的数据和业务逻辑。它负责处理数据的存储、检索、更新和验证。模型不依赖于视图或控制器，而是独立于它们存在。它提供了对数据的访问接口，以供控制器使用，并通知视图进行更新。

视图（View）：视图是用户界面的呈现层。它负责展示模型中的数据给用户，并接收用户的输入。视图通常是由模板或页面组成，可以是HTML、XML、JSON等格式。它不直接处理数据，而是从模型中获取数据并将其显示给用户。

控制器（Controller）：控制器是模型和视图之间的协调者。它接收来自用户界面的请求，并根据请求的类型选择相应的操作。控制器从模型中获取数据，并将其传递给视图进行显示。此外，控制器还可以根据需要更新模型中的数据。总之，控制器负责处理用户输入、协调模型和视图之间的交互。

1. 简述什么是CSS以及CSS的作用

CSS（层叠样式表，Cascading Style Sheets）是一种用于描述网页上元素样式的标记语言。它与HTML结合使用，可以控制网页的布局、字体、颜色、背景等方面的外观。

CSS的作用主要包括以下几个方面：

样式定义：通过CSS，可以为HTML文档中的各个元素指定不同的样式和外观效果。例如，可以设置文字的颜色、字体、大小，调整边距和填充等。

分离样式和内容：CSS使得样式和内容分离，将网页的结构和样式分开管理。这样，在更改样式时，只需修改CSS文件，而不需要修改HTML文档的结构。这提高了代码的可维护性和重用性。

布局控制：CSS提供了丰富的布局功能，允许开发者以灵活的方式控制网页元素在页面中的位置和大小。通过设置盒模型、浮动、定位等属性，可以实现多栏布局、响应式设计等复杂的页面布局效果。

响应式设计：CSS还支持响应式设计，使网页能够适应不同的设备和屏幕尺寸。通过使用媒体查询和弹性布局等技术，可以根据用户的设备自动调整网页的布局和样式，提供更好的用户体验。

提高性能：CSS的加载速度比图片或JavaScript等资源要快得多。通过将样式信息集中在一个CSS文件中，并使用浏览器缓存，可以减少页面的加载时间，提高网站的性能。

1. 简述什么是内置对象，以及内置对象的作用？

内置对象是指由Web容器（比如Java的Servlet容器或JavaScript的浏览器环境）提供的一组对象。这些对象可以直接在应用程序的代码中使用，无需额外的实例化或导入。

内置对象的作用是为开发者提供方便的访问和操作，以处理与Web应用程序相关的功能和信息。它们通常具有特定的属性和方法，用于执行特定的任务。下面是几个常见的内置对象及其作用：

请求对象（Request Object）：用于获取客户端发送的HTTP请求信息，如请求参数、请求头、请求方法等。开发者可以从请求对象中获取和处理用户提交的表单数据、URL参数等。

响应对象（Response Object）：用于生成和发送HTTP响应给客户端。开发者可以通过响应对象设置响应状态码、响应头、响应内容等，以向客户端返回所需的数据。

会话对象（Session Object）：用于在不同页面之间共享数据。会话对象允许开发者存储和检索用户特定的信息，如登录状态、购物车内容等。它通过为每个用户分配唯一的会话ID，在服务器端跟踪用户的会话状态。

上下文对象（Context Object）：提供对Web应用程序上下文的访问。上下文对象包含了关于当前应用程序的信息，如应用程序名称、部署环境等。开发者可以使用上下文对象获取全局配置信息或执行与应用程序上下文相关的操作。

定时器对象（Timer Object）：在JavaScript中，定时器对象用于实现延迟执行和周期性执行的任务。开发者可以使用定时器对象来调度函数、动画效果或其他需要定时触发的操作。

1. 请简述在ideal中创建JSP页面的步骤

打开IDE：启动您选择的IDE，如Eclipse、IntelliJ IDEA等。

创建项目：在IDE中，创建一个新的Java Web项目。这通常涉及选择适当的项目模板和指定项目的名称和位置。

创建JSP文件：在项目中的Web内容目录（如WebContent或src/main/webapp）中创建一个新的JSP文件。您可以通过右键单击相应的目录并选择"New"（或类似选项），然后选择"JSP"（或"Web"）来创建JSP文件。

编写JSP代码：使用IDE的编辑器，在新创建的JSP文件中编写JSP代码。JSP代码可以包含HTML标记和嵌入的Java代码，用于动态生成网页内容。

配置部署描述符：在WEB-INF目录下的web.xml文件中，配置JSP文件的URL映射。这将确定JSP文件的访问路径。

运行和测试：保存并运行项目，以便访问和测试您的JSP页面。在浏览器中输入相应的URL，应该能够看到JSP页面的输出结果。

1. 简述POST请求和GET请求有什么不同？（至少2点）

数据传输方式：GET请求使用URL参数将数据附加到请求的URL上，而POST请求通过请求正文（Request Body）将数据发送给服务器。GET请求的数据会显示在URL中，可以被缓存、书签等直接可见；而POST请求的数据不会显示在URL中，对于敏感信息的传输更安全。

数据大小限制：GET请求对数据大小有一定的限制，因为URL的长度是有限制的，通常约为2048个字符。超过这个限制可能导致URL截断或请求失败。而POST请求则没有明确的大小限制，因为数据是通过请求正文传输的，可以处理大量的数据。

安全性：GET请求的数据暴露在URL中，容易被第三方获取、篡改或伪造。因此，GET请求适合传输无关安全性的非敏感数据。POST请求的数据在请求正文中，相对更安全，不容易被拦截或修改。因此，POST请求适合传输敏感信息，如用户密码、支付信息等。

缓存机制：GET请求具有缓存机制，浏览器会缓存响应结果，下次再请求相同URL时，如果缓存有效，则直接从缓存中获取结果，减少服务器压力。POST请求默认不缓存结果，每次请求都会向服务器发送请求。

56. 请按照以下要求设计一个能处理以get方式提交表单，并包含（username=张三&password=123&hobby=游戏&hobby=踢球）的类ChineseServlet。

要求如下：

创建一个ChineseServlet类，使其继承HttpServlet类并重写该类的doGet()方法。

在doGet()方法中接收参数并解决乱码，并进行输出到控制台。doPost()调用doGet()进行类似处理。

import java.io.IOException; import javax.servlet.ServletException; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse; public class ChineseServlet extends HttpServlet { protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { request.setCharacterEncoding("UTF-8"); // 设置请求编码为UTF-8 // 获取表单参数 String username = request.getParameter("username"); String password = request.getParameter("password"); String[] hobbies = request.getParameterValues("hobby"); // 输出到控制台 System.out.println("Username: " + username); System.out.println("Password: " + password); if (hobbies != null) { System.out.print("Hobbies: "); for (String hobby : hobbies) { System.out.print(hobby + " "); } System.out.println(); } // 可选：返回响应给客户端 response.setContentType("text/html; charset=UTF-8"); // 设置响应类型和编码 response.getWriter().println("Received GET request successfully!"); } protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { doGet(request, response); // POST请求调用doGet()方法进行处理 } }