

Javaweb 复习题

1. 什么是超文本协议？一般 JSP 应用程序服务器（WEB 服务器）使用的发布端口是多少？
超文本协议（HTTP）Hypertext Transfer Protocol:进行超文本和超媒体数据的传输。
发布端口为 8080
2. Eclipse 一般通过什么来进行功能的扩展。插件
3. JSP 有哪三种脚本元素
JSP 中的脚本元素主要包括三个主要部分：
--脚本段：<%This is a scriptlet%>
--声明：<%!This is a declaration%>
--表达式：<%=This is a expression%>

4. JSP 的内置对象有哪九个？

No.	内置对象	类型
1	<i>pageContext</i>	<i>javax.servlet.jsp.PageContext</i>
2	<i>request</i>	<i>javax.servlet.http.HttpServletRequest</i>
3	<i>response</i>	<i>javax.servlet.http.HttpServletResponse</i>
4	<i>session</i>	<i>javax.servlet.http.HttpSession</i>
5	<i>application</i>	<i>javax.servlet.ServletContext</i>
6	<i>config</i>	<i>javax.servlet.ServletConfig</i>
7	<i>out</i>	<i>javax.servlet.jsp.JspWriter</i>
8	<i>page</i>	<i>javax.lang.Object</i>
9	<i>exception</i>	<i>java.lang.Throwable</i>

以上的 9 个内置对象中比较常用的是 pageContext、request、response、session、application 这 5 个。（有关内置对象的使用详细看一下课本或 PPT）

5. WEB-INF 文件夹下进行 SERVLET 配置的文件是什么？web.xml

6. JSP 中 request 对象采用什么方法获取表单提交信息？

单一的参数都可以使用 `getParameter()` 接收，而一组参数（同名参数）要用 `getParameterValues()` 接收。

在表单控件中，像文本框(text)、单选按钮(radius)、密码框(password)、隐藏域(hidden)等，一般都会使用 `getParameter()` 方法进行接收，因为这些控件在使用时参数的名称都只有一个不会重复；而像复选框 (checked)，一般参数的名称都是重复的，是一组参数，所以只能使用 `getParameterValues()` 方法接收，如果不小心使用了 `getParameter()` 方法，则只会接收第一个选中的内容。

7. 客户端向服务端提交数据的方法有哪两种？

get 请求、post 请求

8. 服务器默认的编码为什么？常见的中文编码为什么？

服务器默认的编码：ISO8859-1

常见的中文编码：GBK

```
<%@ page contentType="text/html;charset=GBK"%>
```

```
<%
```

```
    request.setCharacterEncoding("GBK") ;    // 按中文接收
```

```
    String str = request.getParameter("info") ; // 接收表单参数
```

```
%>
```

```
<h2>参数内容：<%=str%></h2>
```

9. JSP 一般通过什么连接数据库，数据库连接类是什么？什么是 SQL 语句类，什么是查询结果集类。

JSP 通过 JDBC（Java DataBase Connectivity 、Java 数据库连接技术的简称）技术操作数据库

查询结果集类 ResultSet

10. SERVLET 本身是个 JAVA 类，必须放置在那个文件夹下？

Servlet 本身就是个 Java 类，所以必须放在 WEB-INF/classes 子文件夹下

11. HTML 语言中的各种常用标签有哪些？

<html></html>、<head></head> 、<title></title>、<meta>、<body></body>等

12. JSP 中加载的基本类有那些？

Import 默认导入的列表是：

java.lang.*

javax.servlet.*

javax.servlet.jsp.*

javax.servlet.http.*

13. 实现客户端和服务端跳转的语句是什么？

服务器跳转:<jsp:forward>

客户端跳转:response.sendRedirect() , (setHeader())

<jsp:forward page= “页面” />

(response.setHeader("refresh","2;URL=ResponseDemo01.jsp") ;// 2 秒后跳转到其他页面)

response.sendRedirect("ResponseDemo01.jsp") ;// 跳转到其他页

<jsp:forward>和 response.sendRedirect() 两种跳转的区别：P151（书上）

①<jsp:forward>属于服务器端跳转，跳转之后地址栏的信息并不会有任何的改变；而 response.sendRedirect()属于客户端跳转，跳转之后地址栏是会改变的，变为跳转之后的页面地址。②在使用 request 属性范围时，只有服务器端跳转才能够将 request 范围的属性保存到跳转页；而如果是客户端跳转，则无法进行 request 属性的传递。③服务器跳转，执行到跳转语句时会立刻进行跳转，而客户端跳转，则是在整个页面执行完之后才执行跳转。

14. JAVA WEB 发布目录中，. jar 放置在那个文件夹中？lib

15. Servlet 的生命周期分为哪几个阶段？

加载 servlet、初始化、处理服务、销毁、卸载（详情见 P257-P258）

16. 编写 JAVABEAN 必须满足哪五点规范？

编写 JavaBean 必须满足以下几点：

①所有的 JavaBean 必须放在一个包中，在 web 中没有包的类是不存在的；

②必须使用 public class 声明 （文件名称与类名称一致），这样才能被外部所访问；

③类中全部属性必须封装，即使用 private 声明；

④封装的属性如果需要被外部所操作，则必须编写对应的 setter、getter 方法。（被封装的属性需要通过 setter、getter 方法设置和取得）

⑤一会 javabean 中至少存在一个无参构造方法，此方法为 JSP 中的标签所使用

Java 命名规范：

包命名：全部字母小写 cn.edu.tjpu

类命名：单词的首字母大写 SimpleBean

属性名称：第一个单词的首字母小写，之后每个单词的首字母大写 studentName

方法命名：和属性命名相同 public void sayHello()

常量命名：全部单词大写 final String DBDRIVER= “TJPU”

17. JSP 中 MODEL2 的工作原理是什么？

Model 2: JSP+JavaBean+Servlet 开发

Model 2 是 MVC 设计模式的一种具体应用

Model 2 中采用 Servlet 作为控制器(Controller)，负责接收客户端 Web 浏览器发送来的所有请求，并依据处理的不同结果，转发到对应的 JSP 页面(Viewer)，实现在浏览器端的显示。

通常 Servlet 中只包含了控制逻辑和一些简单的处理逻辑，更加复杂的业务处理逻辑则借助特定的 JavaBean 具体实现，如：实现与数据的连接，对数据库的数据信息维护修改等。

18. 过滤器的应用有那些，简要说明？（P283）

19. 同名参数是如何接收的？

如果想要接收多个参数，则必须使用：

Public String[] getParameterValues(String name) 方法

此方法返回的是一个数组，表示接收同名的一组参数。

20. **熟悉登录程序的实现！**

21. MVC 分别是指的什么？

MVC (Model-View-Controller) 模型：

模型 (Model)：业务逻辑层，实现具体的业务逻辑、状态管理的功能

视图 (View)：表示层，即与用户实现交互的界面，通常实现数据的输入和输出功能

控制器 (Controller)：控制层。起到控制整个业务流程 (Flow Control) 的作用，实现 View 和 Model 部分的协同工作

22. 获得客户端 IP 地址应该使用什么对象？

直接使用 request 对象的 getRemoteAddr() 方法就可以取得客户端的 IP 地址。

23. JSP 页面输出通常使用哪两种语句？如何书写

一共学习了两种输出：

--out.print ()

--<%=%>


```

<%
    out.println("<table>");
    for(int i=0;i<10;i++){
        out.println("<tr>");
        for(int j=1;j<10;j++){
            out.println("<td>"+(i*j)+"</td>");
        }
        out.println("</tr>");
    }
    out.println("</table>");
%>

```

<%--使用<%=>进行输出-->

```

<table border="1">
<%
    for(int i=0;i<10;i++){
%>
        <tr>
<%
            for(int j=1;j<10;j++){
%>
                <td><%=i*j%></td>
<%
            }
%>
        </tr>
<%
    }
%>
</table>

```

从代码中可以清楚的发现，HTML代码和JAVA代码合理的分割开了

24. GET 方法和 POST 方法的区别是什么？

get 提交

会在地址栏中显示提交的内容，
地址栏中最大能放 4K~5K 的文字，所有过大肯定就无法提交。

post 提交

不在地址栏中显示提交的内容；
本身不受长度的限制，理论上是任意值，但是如果上传内容过多，则会超过 HTTP 的超时时间的限制。

25. JSP 中两种包含方式的区别是什么（<jsp:include page="relativeURL" />和<%@include file="relativeURL" %>的区别。）？


```
<%@ include file="relativeURL" %>
```

此种包含属于静态包含，可以包含任意文件，当然只是把文件内容包含进来。

```
<jsp:include page="relativeURL" />
```

此语句为动态包含，如果被包含的页面是 JSP，则先处理之后再将其结果包含，而如果包含的是非*.jsp 文件，则只是把文件内容静态包含进来，功能与@include 相似。

此指令语法使用非常严格，有以下两种形式：

语法 1:

```
<jsp:include page="页面" />
```

语法 2:

```
<jsp:include page="页面" >
```

```
    <jsp:param name="参数名称" value="值" />
```

.....

```
</jsp:include>
```

两种包含方式的区别：

动态包含的运行结果显示正常，说明动态包含是先将各个页面先处理，处理完后再把内容包含进来。

比较之后可以发现动态包含比静态包含更方便，并且可以像包含文件传递参数。

26. JDBC-ODBC 桥方式如何连接数据库？

27. 能够通过书写程序方式连接 MYSQL 数据库，执行查询和修改的操作。

下面的代码仅供参考：

```
<head>
  <title>JSP读取数据库</title>
</head>

<body>
  <table border="1" align="center">
    <tr>
      <th>书号</th>
      <th>书名</th>
      <th>作者</th>
      <th>出版社</th>
    </tr>
    <%
      String driverClass="com.mysql.jdbc.Driver";
      String url="jdbc:mysql://localhost:3306/simple";//之前数据库名simple
      String user="root";//MySQL帐号
      String password="root";//MYSQL密码
      Connection conn;
      try {
        Class.forName(driverClass).newInstance();
        conn=DriverManager.getConnection(url,user,password);
        Statement stmt=conn.createStatement();
        String sql="SELECT * FROM book";//表名book
        ResultSet rs=stmt.executeQuery(sql);
        while(rs.next())
        {
          %>
```

```
<tr>
  <td><%=rs.getString("bookid")%></td>
  <td><%=rs.getString("bookname")%></td>
  <td><%=rs.getString("writer")%></td>
  <td><%=rs.getString("publish")%></td>
</tr>
<%
  }
  }catch(Exception ex){
    ex.printStackTrace();
  }
%>
</body>
</html>
```