

//每个题2分

1.<jsp:include page="/b.jsp" /> //动态包含, 运行时的事情, 可以包含程序

<%@ include file="b.jsp" %> //静态包含, 编译时的事情, 发生在jsp转换为java 文件中, 更快

2./\* 7.jsp隐含对象 implicit objective \*/

//jsp语言与定义了9个隐含的对象(在jsp中无需创建就可以使用的对象)

描述                      对象

HttpServletRequest request

HttpServletResponse response

HttpSession session //

ServletConfig config //

ServletContext application //考

JspWriter out

PageContext pageContext //回去看PageContext.pageContext对象是PageContext类的实例

page this

Exception exception

---

//PageContext \*\*\*\*\* extends JspContext \*\*\*\* 包 javax.servlet.jsp.JspContext

pageContext对象代表了该jsp页面的运行环境。

(1)在pageContext中有很多获得其他隐式对象的方法

getSession()

getOut()

getPage()

getRequest()

getResponse()

getServletConfig()

getServletContext()

getException()

forward(...)

include(...)

(2)pageContext能作为一个域对象来存储数据, 通常称为page域, 而这个page域的范围只是在当前JSP页面中。

//pageContext.setAttribute(String name,Object value)

//pageContext.getAttribute(String name)

JSP页面中有一个EL表达式:\${data}

在Servlet中就会被翻译成 pageContext.findAttribute("data")。 //全域查找

//从4个域中寻找存储的数据, 查找的顺序由小到达 (page->request->session->context)

---

ServletContext是一个全局的存储信息的空间, 服务器开始就存在, 服务器关闭才是放

一个jsp文件有且仅有一个application对象

//ServletContext:

Object getAttribute(String name)

void setAttribute(String name,Object object)

Enumeration<String> getAttributeNames()

addFilter(...),addListener(...),addServlet(...)

ServletContext getContext(String uripath)

3. <tag-class> 调用哪个类的doEndTag()函数 \*\*\*考

4. <%@ taglib uri="/WEB-INF/a.tld" prefix="" %> //更好  
<%@ taglib uri="与.tld文件中的一样" prefix="a" %> \*\*考试考

```
5. public void doTag()throws JspException,IOException{

    SimpleDateFormat sdf = new SimpleDateFormat("hh:mm");
    String str = sdf.format(new java.util.Date());
    getJspContext().getOut().print(str); //考函数名称PageContext**
}
```

6. <?xml version="1.0" encoding="utf-8" ?>

<!DOCTYPE taglib PUBLIC "">

<taglib>  
 <uri>http://java.sum.com/jsp/jstl/core</uri> /\*\*考,卸载下面的<%@ taglib uri=""里

```
    <tag>
        <name>time</name>
        <tag-class>TimeTag</tag-class>
        <tei-class>MyTagExtra</tri-class> //考
        <body-content>JSP</body-content>
        <attribute> //TimeTag类里的成员变量
            <name>format</name>
            <type>java.lang.String</type>
            <required>false</required>
            <rtexprvalue>true</rtexprvalue> //可动态
        </attribute>
    </tag>
</taglib>
```

7. BodyTag

//迭代 IterationTag

返回值\*\*\*考

EVAL\_BODY\_AGAIN重新解析体内容

SKIN\_BODY

8. 前端控制器根据url地址找到后端控制器 \*\*考

9. struts的配置文件struts-config.html(struts下的配置文件 不一定叫这个) \*\*考  
j2EE的配置文件一定是web.html,但是web.html不是必须的 \*\*考

10. 一切后端控制器的父类是Action \*\*考

11. 对于一个struts 的Action 一定需要一个formBean吗? \*\*考 --不一定, formBean不是必须的

12. 浏览器地址栏: url部分? 名=值 request.getParam()获得值 request.getUri() 获得url部分 \*\*考

13. httpServlet sendRedirect()考 浏览器地址栏变化

和forward() 浏览器地址栏不变

#### 14. ServletContext和ServletConfig(4分)

---

#### 15. JavaEE 访问数据库的程序 (填空)

```
import java.sql.*
import javax.sql.* //填空, 填这个

ServletContextListener    HttpSessionEvent **考HttpSessionListener 这俩Listener
public void contextInitialized(ServletContextEvent sce){ //****考试考 填空
    DataSource ds;

    sce.getServletContext().setAttribute("ds",ds);

    Connection con = ds.getConnection();

    con.close();
}

public class TextJdbc{

    public void test(){

        //DataSource在包import.javax.sql.*中 *****考试考
        DataSource ds = new BasicDataSource(); //在Web应用程序启动是就创建

        //ds.setUrl();
        //ds.setUserName();

        //加载驱动程序
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = //jdbc直连, 创建数据库连接对象 参数(连接数据库的jdbc url, 用户名, 密码)
            DriverManager.getConnection("jdbc:mysql://localhost,"java","java1234");
        //statement,PreparedStatement

        PreparedStatement pstmt =
            con.prepareStatement("select * from student where javaee=? and gender=1");
        //直连不好, 用户特别多时, 多少个用户多少个链接, 慢, 效率低。应该使用链接池

        pstmt.setInt(1,60);
        pstmt.setInt(2,2);

        ResultSet rs = pstmt.executeQuery(); //结果集
        while(rs.next()){
            int id = getInt(1);
            int score = getInt(2);
            System.out.print("id" + id + "score" + score );
        }

        pstmt.close();
        rs.close();
    }
}
```

```

        con.close();
    }
}

```

## 16.jstl通用标签 考试考了

out标记, set标记,

(1) `<c:out>`标签用来显示一个表达式的结果, 与`<%= %>`作用相似, 它们的区别就是`<c:out>`标签可以直接通过`."`操作符来访问属性。如果想要访问`customer.address.street`, 只需写: `<c:out value="customer.address.street">`。

//value:要输出的内容, default:输出的默认值,

```
<c:out value="<string>" default="<string>" escapeXml="<true|false>"/>
```

(2) `<c:set>`设置变量值和对象属性

```
<c:set var="<string>" value="<string>" target="<string>" property="<string>" scope="<string>" />
```

流程控制: : 考了if标记和when标记

(1)`<c:if>`标签判断表达式的值, 如果表达式的值为 `true` 则执行其主体内容。

```
<c:if test="<boolean>" var="<string>" scope="<string>">
```

```
...
```

```
</c:if>
```

(2)`<c:choose>`

```
<c:when ... >
```

```
...
```

```
</c:when>
```

```
</c:choose>
```

\*\*\*\*\*例子

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```
<html>
```

```
<head>
```

```
<title>c:choose 标签实例</title>
```

```
</head>
```

```
<body>
```

```
<c:set var="salary" scope="session" value="${2000*2}"/>
```

```
<p>你的工资为 : <c:out value="${salary}"/></p>
```

```
<c:choose>
```

```
<c:when test="${salary <= 0}">
```

```
    太惨了。
```

```
</c:when>
```

```
<c:when test="${salary > 1000}">
```

```
    不错的薪水, 还能生活。
```

```
</c:when>
```

```
<c:otherwise>
```

```
    什么都没有。
```

```
</c:otherwise>
```

```
</c:choose>
```

```
</body>
```

```
</html>
```

\*\*\*\*\*结果

你的工资为 : 4000

不错的薪水，还能生活。

#### 17. 参数游戏程序 (填空)

```
public class NumberGuessServlet extends HttpServlet{ //考试考

    public void service(HttpServletRequest request
                        HttpServletResponse response)
        throws ServletException,IOException{
        NumberGuessBean bean = null;

        HttpSession session = request.getSession();
        Object o = session.getAttribute("bean");
        if(o == null){
            bean = new NumberGuessBean();
            session.setAttribute("bean",bean);
        }
        else{
            bean = (NumberGuessBean) o;
        }

        String guess = request.getParameter("guess");
        //HttpServletRequest 函数重点考, HttpServletResponse函数

        if(guess == null || guess.length() == 0){
            bean.setGuess(guess);
        }

        if(bean.getSuccess()){
            request.getRequestDispatcher("/success.jsp").forward(request,response);
            bean.reset(); //把参数游戏复位
            //out.println("Congratulation!,you have made " + bean.getCount());
        }
        else if(bean.getCount()==0){
            request.getRequestDispatcher("/welcom.jsp").forward(request,response);
            //out.println();
        }
        else{

        }
    }
}

<%@ page import = "NumberGuessBean" %>
<jsp:usebean id="bean" class="NumberGuessBean" scope="session" />

Congratulation! you've made <jsp:getProperty name="bean" property="numGuess" />
care to <a href="numguess">try again?</a>
```