**JAVAWeb学习笔记**

1. Tomcat的端口配置在该文件下的server.xml文件中，修改TomCat的端口号在Connector中
2. Jsp的主要作用是代替servlet程序回传HTML页面的数据
3. File协议表示浏览器直接读取file:协议后面的路径，解析展示在浏览器上即可

Jsp的page属性或行为 <%@ **page** contentType="text/html;charset=UTF-8" language="java" %>

1. Language 属性，表示jsp翻译过后是什么语言类型，暂时只支持java ②contentType表示jsp返回的数据类型是什么，也是源码中response.setContentType（）参数值③pageEncoding属性，表示当前jsp页面文件本身的字符集④import跟源码中一样，用于导包，导类 ⑤autoFlush属性，设置out输出缓冲区满了之后，是否自动刷新冲级区，默认值是true⑥buffer属性，设置缓冲区的大小，默认值是8kb⑦errorPage设置当jsp页面运行出错时，自动跳转去的错误页面路径⑧isErrorPage属性，设置当前jsp页面是否是错误信息页面 ，默认是false，如果是true可以获取异常信息 ⑨session属性，设置当前jsp页面，是否会创建HttpSession对象。默认是true⑩extends属性，设置jsp翻译出来的java类默认继承谁
2. <%! %>生成脚本 <%=表达式%>表达式脚本 <%java语句%代码脚本：
3. Jsp的三种注释：1：html注释：html注释会被翻译到\_jspService方法里，以out.write（）输出到客户端2：Java注释：Java注释会被翻译到源代码中3：jsp注释<%--这是jsp注释--%>jsp注释可以注掉所有代码
4. Jsp的九大内置对象：①request 请求对象②response 响应对象③pageContext jsp的上下文对象④session 会话对象⑤ application ServletContext对象⑥config ServletConfig对象⑦out 输出流对象⑧page 指向当前jsp的对象⑨exception异常对象
5. Jsp四大域对象：①pageContext （pageContextImpl类）当前jsp范围内有效②request （HttpServletRequest类） 一次请求内有效③ session （HttpSession类）一次会话范围内有效 ④application （ServletContext类）整个web工程范围内都有效（只要web

工程不停止，数据都在） 四个域在使用的时候，优先顺序分别是，他们从小到大的顺序pageContext->request->session->application

1. Out.write()输出字符串没问题，out.print()输出任何数据都没有问题（全部转换成字符串后调用的write输出）
3. EL表达式中的11个隐含对象 ：1：pageContext 类型：pageContextImpl 作用：他可以获取jsp中的九大内置对象2：pageScore 类型：Map<String,OBjectt>他可以获取pageContext域中的数据3：requestScore 类型：；Map<String，Object> 可以获取Request域中的数据 4：SessionScore
4. 类型： Map<Stirng,Object> 可以获取Session域中的数据 5：applicationScore 类型：Map《String，Object》 获取servletContext域中的数据 6：param 类型：Map《String，String》 他可以获取请求参数的值 7：paramValues类型：Map<String,String[]> 他也可以获取请求参数的值，获取多个值的时候使用 8：header 类型： Map<String,String> 获取请求头的值 9headerValues 类型：Map<String，String[]> 获取请求头的值，可以获取多个值的情况 10：cookie 类型：Map<String ,Cookie> 获取当前请求的Cookie信息 11：initParam 类型： Map<String,String> 获取在web.xml中配置的<context-param上下文参数
5. JSTL：1：<c:set/>标签：可以往域中保存数据 2：<C:if/>标签用来判断 3：

<c:choose><c:when><c:otherwise>标签 用来多路判断。switch…case…default非常接近 3：<c:forEach/>标签 遍历输出使用

1. 文件的上传介绍：1.要有一个form标签，method=post请求2：form标签encType属性值必须是multipart/form-data值 表示提交的数据以多段（每一个表单项，一个数据段）的形式进行拼接，然后以二进制流的形式发送给服务器 3：在form标签中使用input type=file添加上传的文件 4：编写服务器代码接收，处理上传的数据

文件下载的方法protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
// 获取要下载的文件名  
 String downoadFileName="2.jpg";  
// 读取要下载的文件内容（通过servletContext对象获取）  
 ServletContext servletContext=getServletContext();  
 InputStream resourceAsStream=servletContext.getResourceAsStream("file/"+downoadFileName);  
// 获取响应的输出流  
 OutputStream outputStream=resp.getOutputStream();  
 IOUtils.*copy*(resourceAsStream,outputStream);  
// 告诉客户端返回的类型是什么  
 String mimeType = servletContext.getMimeType("file/"+downoadFileName);  
 resp.setContentType(mimeType);  
// Content-Disposition 表示响应头，表示收到的数据怎么处理  
// attachment表示附件，表示下载使用  
// filename表示要下载的文件名  
// url编码，解决乱码问题，支持谷歌浏览器和ie浏览器  
// resp.setHeader("Content-Disposition","attachment;filename="+ URLEncoder.encode("中国.jpg","UTF-8"));  
 resp.setHeader("Content-Disposition","attachment;filename==?UTF-8?B"+new BASE64Encoder().encode("中国.jpg".getBytes(StandardCharsets.*UTF\_8*)));  
  
  
  
 }

1. BeanUtils工具类可以一次性的把所有请求的参数注入到javaBean中
2. Cookie是服务器保存键值对的一种技术