# day21 JSP

## JSP简介

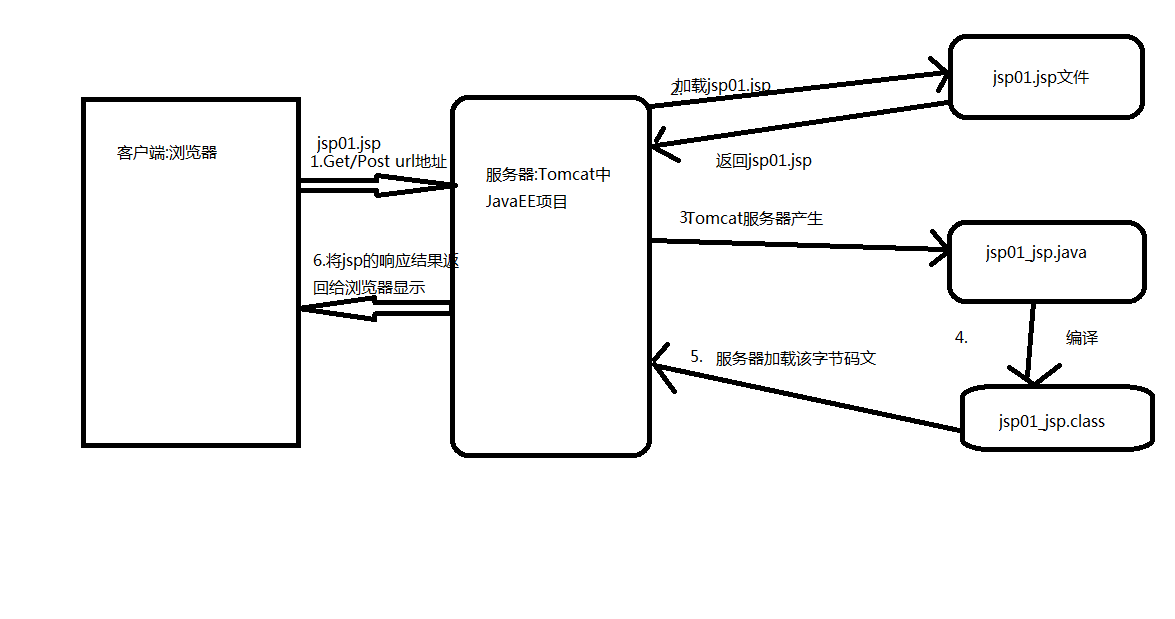
JSP:Java Server Page,是实现动态网页开发的一种技术.通常JSP就是在普通的HTML页面嵌入Java代码.JSP本质上其实就是Servlet.我们主要是通过JSP来实现Web程序的页面部分.

通常是在ＨＴＭＬ中，通过<%%>这个标记在HTML中嵌入Java代码.

## JSP的优势

1. 相对于ASP来说,JSP使用Java来编写的,可以实现跨平台使用;
2. 与纯粹的Servlet相比,Servlet也可以输出Html页面,但是如果是复杂的页面,Servlet实现Html会特别复杂.JSP实现HTML页面明显简单太多了.
3. 与JavaScript相比,JavaScript虽然也能与Html交互,但是它很难与服务器进行交互,而JSP既可以与HTML交互,还可以很容易的与服务器交互.
4. 与纯粹的HMTL页面相比,html页面只能是静态的页面,无法实现动态效果.

## JSP执行流程(原理)



## 四.JSP的具体使用

### 1.JSP输出方式

方式一:



方式二:



方式三:



### 2.JSP中的注释

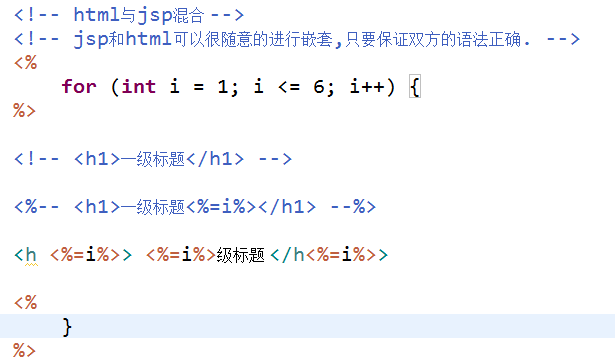
①.普通的html注释:



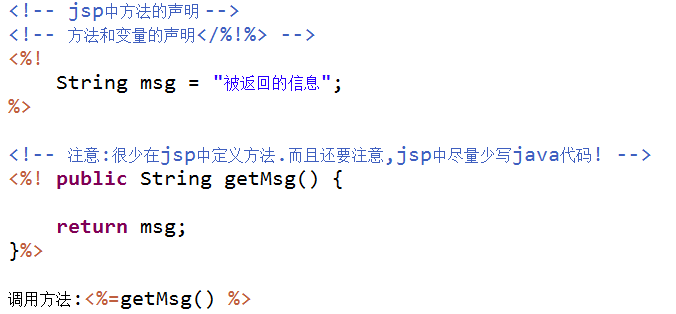
②.JSP自己的注释:



### 3.JSP与HTML混合



### 4.JSP中方法与变量的声明,调用



## 五.JSP的指令

### 1.page指令

language=*"java"*

contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*

import=*"java.\*,java.util.\*"*

buffer=*"8kb"*

autoFlush=*"true"*

isELIgnored=*"false"*

isThreadSafe=*"true"*

session=*"true"*

language:指定jsp的服务端的编程语言是java,只能是Java.

contentType:指定jsp页面的mimetype类型和页面的编码;

pageEncoding:指定jsp页面中的字符编码;

import:导包的属性;

buffer:指定jsp页面的缓冲区大小,默认是8kb;

autoFlush:自动刷新jsp页面缓冲区里的内容.但是如果buffer的值是0kb或者none,autoFlush就无效了.

isELIgnored:是否忽略EL表达式,一般都不忽略,默认值是false.

isThreadSafe:指定当前的jsp页面是否是线程安全的.

session:是否支持session功能,默认为true,否则session对象就无法获取;

errorPage:引用一个错误处理页面;

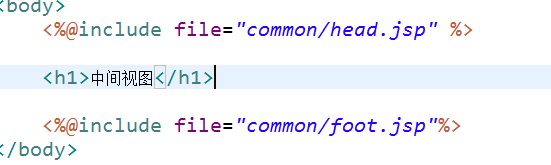
isErrorPage:指定当前页面是否是错误处理页面,默认是false.

注意:

isErrorPage应该为true,exception对象才能使用.

### 2.include指令

include用来引用其他的html或者jsp页面.



### 3.taglib指令

运行用户进行自定义标签的指令.

## 六.jsp的缓冲区

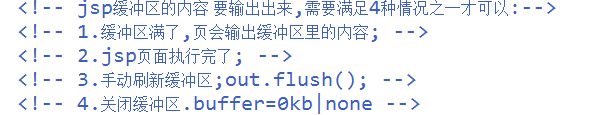
### 1.jsp缓冲区,默认大小8kb



如果jsp页面内容非常少,不用缓冲区效率更高一些;

如果jsp页面内容非常多,使用缓冲区效率更高一些.

### 2.jsp缓冲里的内容输出出来满足的4个条件(重要):



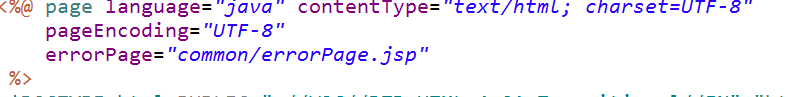
### 3.out.write()与response.getWriter().write()的区别:

out是JspWriter类的对象,JspWriter类自定缓冲功能;

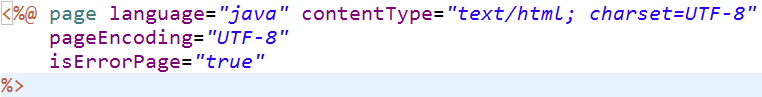
response.getWriter()类的对象是PrintWriter,是不带缓冲功能的.

## 七.错误页面处理

### 1.引用错误页面:errorPage属性



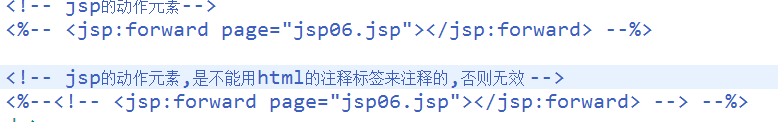
### 2.定义错误页面:isErrorPage



注意:

isErrorPage应该为true,exception对象才能使用.

## 八.JSP的动作元素

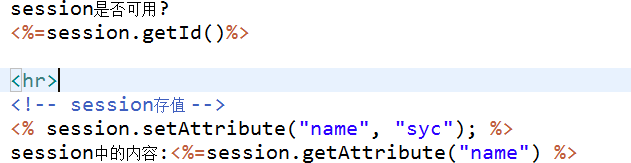


## 九.JSP的Session功能

开启Session功能



Session存值取值:



## 十.JSP的9个内置对象(重要)

### 1.request:

HttpServletRequest---🡪request,代表了用户在浏览器上发送给服务器的请求,给对象里面包含了请求参数(get请求字符串后面的内容,post表单里的内容)的信息.

request.setAttribute(name,value);

request.getAttribute(name);

request.getRequestDispatcher(url).forward(request,response);

### 2.response:

HttpServletResponse--🡪response,代表了服务器给浏览器的响应结果.

response.getInputStream();

response.getWriter().write()|print();

response.addCookie();

### 3.out:

JspWriter类--🡪out,该对象是带有缓存功能的输出对象.

通过利用该对象在浏览器上输出一些内容.

out.write();

out.print();

out.flush();

### 4.session:

HttpSession类---🡪session,session是在服务器保存用户信息的技术,记录用户的会话状态.

session.setAttribute(key,value);

session.getAttribute(key);

session.removeAttribute(key);

session.invalidate();

### 5.exeception:

默认情况下,jsp的java源代码是没有exeception这个对象的.只有当某个jsp中,page指令的isErrorPage属性设置为true的时候,才会创建出来exeception对象.

定义和定位错误信息.

### 6.pageContext:

利用这一个pageContext可以获取其他的8个对象.

PageContext类--🡪pageContext,代表了一个页面上下文.

该对象代表了整个的JSP页面,通过该对象,可以访问整个JSP页面中的信息.

pageContext.setAttribute(name,value,scope);

pageContext.getAttribute(name,scope);

pageContext.removeAttribute(name);

pageContext.forword(url);

pageContext.include(url);

pageContex.getSession();

pageContext.getRequest();

pageContext.getResponse();

....

### 7.page:

Object类--🡪page对象,page=this;

代表了当前类本身.

该对象不是很常用,一般可以利用该对象去调用本类中的一些方法,变量..;

### 8.config:

ServletConfig类---🡪对象,init(ServletConfig config).

代表了Servlet里面的一些配置信息,尤其是web.xml文件的配置信息.

我们可以利用该对象来获取web.xml文件中servlet节点里的

init-param节点里的信息.

config.getInitParam(name);

config.getInitParamNames();

config.getServletName();

### 9.application:

ServletContext类--🡪application,代表了整个的Web项目.

该对象的生命周期非常长,从Web服务器开启开始,一直到服务器停止位置,application生命周期就这么长.

application.setAttribute(name,value);

通常情况下,可以利用application来存储全局的变量,常量等信息,供所有的Servlet来共享.

## 十一.JSP的4个作用域(重要)

### 1.jsp的4个作用域



2.常用方法:

pageContext.setAttribute(name,value,scope);

pageContext.getAttribute(name,scope);

pageContext.removeAttribute(name,scope);

pageContext.findAttribute(name,scope);



取值:

