1. JSP的页面元素：

a. 脚本Scriptlet

1. <%

局部变量，java语句

%>

2. <%!

全局变量，方法

%>

3. <%=输出表达式%>

注意：out.println()不能回车；要想回车：
，可以直接解析Html代码

2. 指令

page指令

<%@ page %>

page指定的属性：

language: jsp页面使用的脚本语言

import:导入类

pageEncoding:jsp自身编码 jsp-->java

contentType:浏览器解析jsp的编码

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

3. 注释

html注释<!---->

java注释 // /* */

jsp的注释 <%----%>

4. JSP九大内置对象(自带的，不需要new也能使用的对象)

out:输出对象，向客户端输出内容

request:请求对象，存储“客户端向服务端发送的请求信息:

常见方法:

String getParameter(String name) 根据key返回value

String[] getParameterValues(String name):根据key，返回多个value（checkbox）

void setCharacterEncoding("编码格式")：设置请求编码(tomcat)

getRequestDispatcher(文件名).forward(request,response);请求转发的方式跳转页面

A -> B

ServletContext getServletContext():获取项目的ServletContext对象

Cookie[] request.getCookies():获取cookies

response:响应对象

常见方法：

void addCookie(Cookie cookie):服务端客户端增加cookie对象

void sendRedirect(String location) throw IOException: 页面跳转的一种方式（重定向）

void setContentType(String type):设置服务器响应端的编码（设置服务端的Content类型）

void setCharacterEncoding(String type):设置响应编码

void addHeader

PrintWriter getWrite():获得out内置对象

ServletOutputStream getOutputStream():获得输出流

pageContext JSP页面容器

session: 会话对象

常见方法：

String getId():获取sessionId

boolean isNew():判断是否是新用户(第一次访问)

void invalidate():使session失效（退出登录，注销）

void setAttribute()

Object getAttribute()

void setMaxInactiveInterval(秒):设置最大有效非活动间隔

int getMaxInactiveInterval(秒)：获取最大有效非活动间隔

application:全局对象

常见方法：

String getContextPath() 虚拟路径

String getRealPath(String name) 绝对路径(虚拟路径 相对的绝对路径)

config: 配置对象(服务器配置信息)

page: 当前JSP页面对象(相当于this)

exception:(异常对象)

5.四种范围对象

pageContext:页面容器 (page对象); 当前页面有效(页面跳转后无效)

request:请求对象 同一次请求有效(重定向无效, 请求转发有效)

session:会话对象 同一次会话有效(无论怎么跳转都有效,关闭/切换浏览器无效,准确说是Cookie失效)只要客户端Cookie中传递的Jsessionid不变,Session不会重新实例化(不超过默认时间).

application:全局对象 全局对象(整个项目有效,切换浏览器仍有效);关闭服物, 其他项目无效

-->多个项目共享,重启后仍然有效: JNDI

共有的方法:

Object getAttribute(String name):根据属性名, 或者属性值

void setAttribute(String name,Object obj):设置属性值(新增, 修改)

void removeAttribute(String name):根据属性名, 删除对象

1.以上四个范围对象, 通过setAttribute()赋值,通过getAttribute()取值

2.以上范围对象尽量使用最小的范围.因为对象的范围越大,造成的性能损耗越大

6.其他对象

Cookie:name=value

javax.servlet,http.Cookie

public Cookie(String name,String value)

String getName():获取name

String get Value(): 获取value

void setMaxAge(int expiry);最大有效期(秒)

服务器准备Cookie：response.addCookie(Cookie cookie)

页面跳转（请求转发，重定向）

客户端获取Cookie：request.getCookies();

7.1.html的form表单提交默认为method="get"和地址栏 请求方式，默认都属于get提交方式(但是地址栏能容纳的信息有限，4-5kb；如果请求大文件，图片等会出现地址栏无法容纳全部的数据)

2.文件操作操作，必须是post

8.统一请求的编码 request

get方式请求 如果出现乱码，解决：

a.统一每一个变量的编码(不推荐)

new String(变量名.getBytes(旧编码，新编码))

name=new String(name.getBytes("iso-8859-1","utf-8"));

b.修改server.xml 一次性的修改

在port后 URIEncoding="UTF-8"(一般不怎么干)

post方式出现乱码

void setCharacterEncoding("编码格式")：设置请求编码(tomcat)

9.重定向和请求转发的区别

	请求转发	重定向
地址栏是否改变	不改变	改变
是否保留第一次请求的数据	保留	不保留
请求的次数	1次	2次
发生跳转的位置	服务端	客户端

10. session和Cookie

Cookie（客户端）：Cookie是由服务端产生，再发送给客户端保存。相当于本地缓存的作用

作用：提高访问服务端的效率，但是安全性较差（密码也会缓存）

除了自己设置的Cookie对象外，还有一个name为JSESSIONID的cookie

建议cookie只保存英文和数字，否则需要进行编码，解码处理

使用cookie实现 记住用户名 功能

session（服务端）：会话（开始-结束）

a.浏览网站:开启-关闭

b.购物：浏览，付款，退出

c.电子邮件：浏览，写邮件，退出

机制：

a.客户端第一次请求服务端时,(JSESSIONID-sessionId)会产生一个

session对象(用于保存客户的信息)

b.并且每个session对象 都会有一个唯一的sessionId(用于区分session)

c.服务端会产生一个cookie，并且该cookie的name=JSESESSIONID,value=

服务端sessionId的值

然后服务端会在响应客户端的同时，将该cookie发送给客户端，至此客户

端就有了一个cookie(JSESESSIONID);因此，客户端的cookie就可以和服

物端的session一一对应（JSESESSIONID--sessionID）

d.客户端第二/n次请求服务端时：服务端会先用客户端cookie中的

JSESESSIONID去服务端的session中匹配sessionId，如果匹配成功，说明

此用户不是第一次访问，无需登录

区别：

	session	cookie
保存的位置	服务端	客户端
安全性	较安全	较不安全

保存的内容	Object	String