**3.JSP隐式对象之输入和输出对象**

* 隐式对象概述
  + JSP隐式对象就是不需要我们new的对象，我们可以直接在JSP页面中使用它们，它们是web容器加载的一组类的对象，隐式对象的名称是JSP的保留字。
  + 输入输出相关的：request，response，out
  + 与作用范围相关的：pageContext，request，session，application
* out对象
  + out对象主要用来向客户端输出各种格式的数据，并且管理应用服务器上的输出缓冲区，out对象的基本类型是javax.servlet.jsp.JspWriter类。
  + out对象最重要的方法如下：out.println(DataType);
* request对象
  + 获取请求参数
    - Web应用是请求/响应架构的应用，浏览器发送请求时可能需要将一些请求参数发送给服务器，服务器端负责解析请求参数的就是JSP或Servlet，而JSP和Servlet取得请求参数的途径就是request。request是HttpServletRequest接口的实例，它提供了如下几个方法来获取请求参数：
    - String getParameter(String paramName)
    - String getParameterValues[](String paramName)
    - Map getParameterMap()
    - Enumeration getParameterNames()
  + 设置request编码的字符集
    - request.setCharacterEncoding("utf-8")
* response对象
  + 使用response对象完成页面跳转
    - void sendRedirect(String name)
  + 重定向常见错误
    - 如果传递给response.sendRedirect()方法的相对URL以"/"开头，它是相对于整个WEB站点的根目录，即此方法可以导致跨应用访问。所以我认为传递给response.sendRedirct()方法的相对URL尽量要写成相对路径或绝对路径。
* 页面之间的跳转
  + 转发（forward）
    - request.getRequestDispatcher().forward()
  + 重定向（redirect）
    - response.sendRedirct()
  + 转发和重定向的区别
    - request.getRequestDispatcher().forward()方法只能将请求转发给同一个WEB应用的组件；而response.sendRedirct()方法还可以重定向到同一个站点上的其他应用程序中的资源，甚至是使用绝对URL重定向到其他站点的资源。如果传递给response.sendRedirect()方法的相对URL以"/"开头，它是相对于整个WEB站点的根目录；如果创建RequestDispatcher对象时指定的相对URL以"/"开头，它是相对于当前WEB程序的根目录。
    - 调用response.sendRedirect()方法重定向的访问过程结束后，浏览器地址栏中显示的URL会发生改变，由初始的URL地址变成重定向的目标URL；调用request.getRequestDispatcher().forward()方法的请求转发过程结束后，浏览器地址栏保持初始的URL地址不变。
    - response.sendRedirct()方法对浏览器的请求直接作出响应，相应的结果就是告诉浏览器去重新发出对另外一个URL的访问请求：request.getRequestDispatcher().forward()方法在服务器端内部将请求转发给另外一个资源，浏览器只知道发出了请求并得到了响应结果，并不知道在服务器程序内部发生了转发行为。
    - request.getRequestDispatcher().forward()方法的调用者与被调用者之间共享相同的request对象和response对象，他们属于同一个访问请求和响应过程；而response.sendRedirct()方法的调用者与被调用者使用各自的request对象和response对象，它们属于两个独立的访问对象和响应过程。