

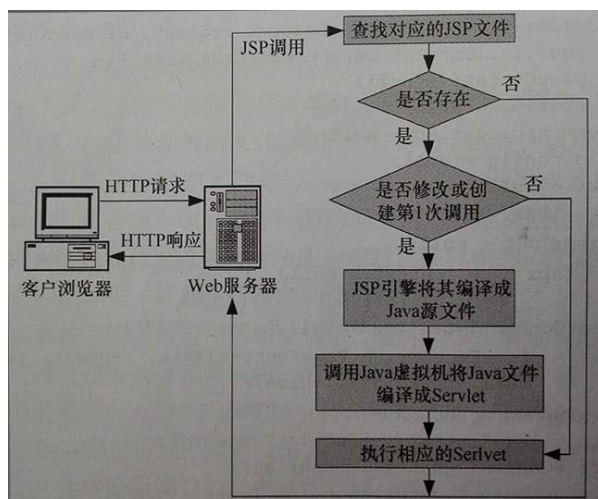
第 3 章 JSP 内置对象

3.1 request 对象（请求）

1. 什么是内置对象？

不用声明和创建就可以在 JSP 程序中使用的成员变量。（JSP 页面中自动创建）

2. request 对象（请求对象，彩礼，生儿生女问题）



图教材 P4

客户端 (男)	====请求====>	服务器端 (女)
	<====响应====	

3. request 对象作用 1：获取表单控件值

(1) 复习表单

```

<form action="跳转页面/执行程序" method="get/post" name="表单名">
  <input type="text" name="控件名 (uname)" />
  <input type="submit" value="ok" />
</form>

```

(2) 前台（JS 客户端）获取控件（uname）的值

```
var txt = 表单名.控件名.value;
```

(3) 后台（JSP 服务器端）获取控件（uname）的值


```
<%= request.getParameter("控件名"); %>
```

(4) getParameter()方法


```
String  getParameter(String)
```

4. request 作用 2: 获取地址栏参数值

```
<a href="b.jsp?username="zhangsan"&password="123456">
```

5. 处理空值 

```
String str = request.getParameter("控件名");  
if(str == null){ str = "";}  
  
=====
```

6. 处理表单提交后中文乱码  (我提供的环境不用写一下语句, utf-8)

(1) 设置 contentType 编码格式为 gb2312, utf-8, gbk

```
<%@page  contentType="MIME;charset=gb2312"%>
```

(2) 获取表单控件值

```
String  uname = request.getParameter("uname");
```

(3) 将控件值解析为字节数组 (教材第 44 页 "ISO8859_1")

```
byte [] b = uname.getBytes("ISO-8859-1");
```

(4) 使用字节数组重新创建字符串 uname

```
uname = new String(b); //uname = new String(b,"gb2312");
```

2.6 include 动作标记

1. 作用: 包含文件 (并传递参数)

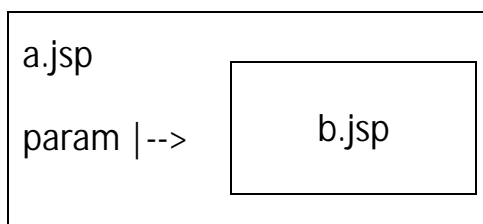
2. 不传参语法格式

```
<jsp:include  page="地址"/>
```


3. 传参语法格式

```
<jsp:include  page="地址" >  
    <jsp:param name="参数名" value="参数值"/>  
</jsp:include>
```

4. 示意图



母体向胎儿传递营养，单方向，当你老了，头发也白了。父母恩不能忘。

5. 传递参数实例  （见“动作标记 - include 实例代码”）

6. include 指令标记和动作标记的区别

①<%@include ~%> ②<jsp:include ~/>

(1) 处理阶段：①编译 (*.java -> *.class) ②运行 (*.class -> 结果)

(2) 处理速度：①快 ②慢


(3) 传递参数：①不可 ②可以

=====

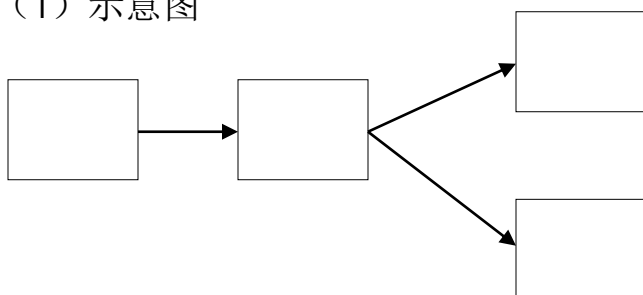
2.7 forward 动作标记

1. 作用：跳转页面

2. 语法格式：<jsp:forward page="相对地址">

3. 登录跳转实例  （见“动作标记 - forward 实例代码”）

(1) 示意图



(2) 参考代码（讲解代码，拷贝代码到文件，运行演示）

登录页面 (login.jsp)

```

<form action="judge.jsp">
  <input type="text" name="uname"/>
  <input type="submit" value="ok"/>
</form>
    
```

判断页面 (judge.jsp)

```
<% String uname = request.getParameter("uname");
    if(uname==null || uname.equals("") || uname.length==0){
        //跳转到 error.jsp
        <jsp:forward page="error.jsp"/>
    }
    if(!uname.equals("zs")){
        //跳转到 error.jsp
        <jsp:forward page="error.jsp"/>
    }
    <% }else{ %>
        //跳转到 success.jsp
        <jsp:forward page="success.jsp"/>
    }
    <% } %>
```

成功页面 (success.jsp)

```
<%//获取表单参数
    String uname = request.getParameter("uname");
    out.print("Welcome,"+ uname);
    %>
```

(4) 错误页面 (error.jsp)

```
<h1>错误</h1>
<a href="login.jsp">重新登录</a>
```

=====

3.1 request 对象 (请求)

7. 在 request 生命周期内设置属性 (键值对)

【理解】

将 request 对象想象成一个表格，第 1 列是属性，第 2 列是属性值。

request 对象（表格）	
属性	属性值
name	“request”
sex	“female”
birth	new Date()

（1）向对象（表格）中添加属性（键和值、设置变量）

```
request.setAttribute(String key,Object value); //void
```

示例代码：

```
request.setAttribute("a",100); //Object a = 100;
```

（2）获取对象（表格）中的属性值（获取键中的值、变量值）

```
request.getAttribute(String key); // Object
```

示例代码：

```
Object a = request. getAttribute ("a");
```

（3）从对象（表格）中移除属性（删除键值对、变量）

```
request.removeAttribute(String key); //void
```

示例代码：

```
request. removeAttribute ("a");
```

（4）实例：在 a.jsp 中设置属性、在 b.jsp 获取属性值


1) a.jsp 页面

```
<% request.setAttribute("a",100);%>
```

```
<jsp:forward page="b.jsp"/>
```

2) b.jsp 页面

```
<% = request.getAttribute("a")%>
```

8. request 获取键值表中所有信息 

（1）设置多个属性

```
request.setAttribute("a",100);
```

```
request.setAttribute("b",200);
```

```
request.setAttribute("c",300);
```

（2）创建枚举对象

【什么是枚举：八卦，64 卦：乾坎艮震、巽离坤兑。金庸武侠小说：飞雪连天射白鹿，笑书神侠倚碧鸳，百家姓】

```
Enumeration e = request.getAttributeNames();
```

(2) 循环获取键值对

```
while(e.hasMoreElements()){
```


(3) 获取键名

```
String key = e.nextElement().toString();
```

(4) 获取键中的值

```
String value = request.getAttribute(key).toString(); }
```

【解释结果】

9. request 获取复选框值 

```
String [] request.getParameterValues("控件名称");
```

(1) 复选框（考试系统的多选题）

```
<input type="checkbox" name="fxk" value="A"/>A. JDK  
<input type="checkbox" name="fxk" value="B"/>B. TOMCAT  
<input type="checkbox" name="fxk" value="C"/>C. ECLIPSE  
<input type="checkbox" name="fxk" value="D"/>D. SQL SERVER
```

(2) Java 代码

```
<%  
String [] fxk = request.getParameterValues("fxk");  
if(fxk != null){ // 防止空指针异常  
    for(int i=0;i<fxk.length;i++){  
        out.print(fxk[i]);  
    }    }else{ out.print("no choice");}  
%>
```

=====


10.request 对象常用方法（教材 P56）

- (1) `String getProtocol` 返回请求用的协议类型及版本号
- (2) `String getServerName` 返回接受请求的服务器主机名
- (3) `int getServerPort` 返回服务器接受此请求所用的端口号
- (4) `String getRemoteAddr` 返回发送此请求的客户端 IP 地址
- (5) `String getRemoteHost` 返回发送此请求的客户端主机名
- (6) `String getContextPath` 返回上下文路径

3.2 response 对象 (响应)


1. 动态响应 `contentType`

- (1) 语法格式: `response.setContentType(MIME);`

- (2) 实例: 

2. 重定向

- (1) 语法格式: `response.sendRedirect(地址);`

- (2) 实例: 

3. 状态行

4. 状态代码

- (1) 404: 请求的资源不可用 (404 图片)
- (2) 500: 服务器内部发生错误

=====

3.3 session 对象 (会话)

1. 引入 session 对象原因

HTTP 协议无状态: 失忆协议, 又称老年痴呆协议。

(1) HTTP 协议记不住哪个页面被访问过



a.jsp b.jsp c.jsp

a.jsp 被访问过吗? b.jsp 被访问过吗? c.jsp 被访问过吗?

(2) HTTP 协议记不住谁 (哪个客户) 访问过某个页面



a.jsp



老王



老张



老李

老王浏览过 a.jsp 吗? 不知道! 老张浏览过 a.jsp 吗? 不知道! 老李浏览过 a.jsp 吗? 不知道!

(3) session 对象作用

在服务器上标注 (标识) 访问网站的不同的客户, 即在服务器上为每个客户端做互不相同的记号。

服务器 (Tomcat) 给访问当前网站的不同客户端机器分配不同的 id 号 (A, B, C)。

客户 A 第一次访问网站 M, 网站 M 的服务器为客户 A 分配一个 session Id 号。



在这个 id 号没失效之前, 当客户 A 再次访问网站 M, 服务器就能使用 session 识别出客户 A。

【引】每个访问网站 M 的人, 都拿着一个 SESSION ID 号, 让我联想起孙燕姿在《遇见》里唱的, 我听见风来自地铁和人海, 我排着队, 拿着爱的号码牌。每个还没找到另一半的人, 手里都拿着爱的号码牌, 排着队张望, 期待那种遇见你是最美丽的意外的场景。

2. 创建 session 对象 (可直接使用)

```
HttpSession session = request.getSession();
```

3. 获取 session 的 Id 号

```
session.getId();
```


4. 会话对象存储数据

【理解】

将 session 对象想象成一个表格，第 1 列是属性，第 2 列是属性值。

session 对象（表格）	
属性	属性值
name	“session”
sex	“male”
birth	new Date()

(1) setAttribute(key,value): 设置属性及值

(2) getAttribute(key): 获取属性值

(3) removeAttribute(key): 删除属性

(4) getAttributeNames(): 获取 session 对象的所有属性值



```
Enumeration e = session.getAttributeNames();
while(e.hasMoreElements()){
String key = e.nextElement().toString();
String value = session.getAttribute(key).toString(); }
```

5. session 对象应用

(1) 购物车（临时存储）



※教材 P90~P92 简单购物车代码※

(2) 验证用户登录

用户跳过登录页面访问主页，在主页代码中使用 session 对象验证。【过滤网（过滤器）】

```
index.jsp --> Login.jsp
```

```
<%
```

```

    if(session.getAttribute('userName') == null){
        response.sendRedirect("Login.jsp");
    }
else{
    out.print("welcome,"+session.getAttribute("userName"));
}
%>

```

6.session 失效

- (1) 调用 invalidate()方法
- (2) 达到“发呆”时间(30')：记笔记--> 发呆看美女 30 分钟 -->记笔记
- (3) 关闭服务器（应用程序（Tomcat）应用对象）

3.4 application 对象（共享）

1.创建共享对象

```
ServletContext application = request.getServletContext();
```

2. 共享对象存储数据

- (1) setAttribute(key,value): 设置属性及值
- (2) getAttribute(key): 获取属性值
- (3) removeAttribute(key): 删除属性
- (4) getAttributeNames(): 获取 application 对象的所有属性值



```

Enumeration e = application.getAttributeNames();
while(e.hasMoreElements()){
    String key = e.nextElement().toString();
    String value = application.getAttribute(key).toString(); }

```

3.5 pageContext 对象（页面（上下文））

1. 获取不同范围（生命周期）对象

- (1) 表格显示

pageContext 对象获取作用域对象		
获取对象	调用方法	返回类型

request	getRequest()	ServletRequest
session	getSession()	HttpSession
application	getServletContext()	ServletContext

(2) 表格显示

request 对象获取作用域对象		
获取对象	调用方法	返回类型
session	getSession()	HttpSession
application	getServletContext()	ServletContext

(3) 语句表示

(1) request

```
ServletRequest request = pageContext.getRequest();
```

(2) session

```
HttpSession session = pageContext.getSession();
```

(3) application

```
ServletContext app = pageContext.getServletContext();
```

2. 设置属性 (教材 P65)

【理解】

将 pageContext 对象想象成一个表格，第 1 列是属性，第 2 列是属性值。

pageContext 对象 (表格)	
属性	属性值
name	“pageContext”
sex	“female”
birth	new Date()

(1) pageContext.setAttribute(k,v);

(2) pageContext.getRequest().setAttribute(k,v);

(3) pageContext.getSession().setAttribute(k,v);

(4) pageContext.getServletContext().setAttribute(k,v);

3. 获取属性 (教材 P65)

(1) pageContext.getAttribute(k);

(2) `pageContext.getRequest().getAttribute(k);`

(3) `pageContext.getSession().getAttribute(k);`

(4) `pageContext.getServletContext().getAttribute(k);`

4.实例：验证各内置对象设置属性（天上一天，地上一一年）

pageContext 对象		request 对象		session 对象		application 对象	
属性	值	属性	值	属性	值	属性	值
a	1	a	10	a	100	a	1000
b	2	b	20	b	200	b	2000
c	3	c	30	c	300	c	3000

3.6 内置对象的生命周期

1.pageContext（page）及其变量生命周期

页面执行开始 至 页面执行结束：执行时间 t



2.request 对象及其变量生命周期

请求开始 至 请求结束：（3 个页面一次性执行完时间 3t）



$$T_{\text{总}} = T_a + T_b + T_c$$

3.session 对象及其变量生命周期

浏览器打开（用户登录网站）开始 至 浏览器关闭（用户离开网站）结束

$$T_{\text{总}} = T_a + T_{\text{上厕所}} + T_b + T_{\text{看美女}} + T_c$$

（用户在网站中停歇时间（发呆时间）和访问页面执行时间的和）

4.application 对象及其变量生命周期

服务器启动开始 至 服务器关闭结束

5.生命周期比较

$$\text{life}(p) < \text{life}(r) < \text{life}(s) < \text{life}(a)$$

【引 1】可叹生命的短暂，时间的珍贵，珍惜眼下。

【引 2】人生与 A4 纸文档，人生苦短

【实例】适当使用监听器知识点